



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

ANALYZÁTOR SIGNÁLU S FPGA

SIGNAL ANALYZER WITH FPGA

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Václav Kraus

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Pavel Šteffan, Ph.D.

BRNO 2018

Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**

Ústav mikroelektroniky

Student: Bc. Václav Kraus

ID: 138558

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Analyzátor signálu s FPGA

POKYNY PRO VYPRACOVÁNÍ:

V rámci diplomové práce navrhnete a realizujete spektrální analyzátor s FPGA a A/D převodníkem s funkcí záznamu surových vzorků do paměti DDR. K analyzátoru také navrhnete úzkopásmový konvertor s možností demodulace.

Prostudujte možnosti ukládání vzorků signálů do DDR paměti pomocí FPGA obvodu a možnosti přenosu dat pomocí rozhraní USB3.0 z FPGA do počítače, dále prostudujte možnosti výpočtu spektra ze vzorkovaného signálu a možnosti digitální filtrace. K navrženému digitálnímu přijímači napište jednoduchou aplikaci pro řízení přijímače a zobrazení dat v počítači. Ověřte funkčnost navrženého analyzátoru.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 5.2.2018

Termín odevzdání: 22.5.2018

Vedoucí práce: doc. Ing. Pavel Šteffan, Ph.D.

Konzultant: Ing. Michal Kováč, Ph.D.

doc. Ing. Lukáš Fujcik, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cílem této práce je prostudovat možnosti výpočtu spektra, možnosti přenosu dat pomocí rozhraní USB 3.0, možnosti ukládání dat do DDR3 paměti pomocí FPGA. Dále je cílem návrh a realizace spektrálního analyzátoru s funkcí záznamu surových vzorků do DDR paměti rozšířeného o úzkopásmový konvertor za využití hradlových polí. Práce je dělena do dvou částí. První část je teoretická. Zabývá se zmíněnou problematikou. Druhá část obsahuje návrh systému. Výsledkem práce je realizace analyzátoru signálu v čipu FPGA řízeného z počítačové aplikace přes USB 3.0.

KLÍČOVÁ SLOVA

FPGA, USB, AD převodník, FFT, Spektrum, FIR filtr, CIC filtr, Radio, DDC, Demodulace, Paměť RAM, DDR3

ABSTRACT

The aim of this thesis is to study the possibilities of spectrum calculations, as well as data transfer via USB 3.0 and data saving to a DDR3 memory via FPGA. The focus is also on design and realization of a spectral analyzer with a record of samples to DDR memory expanded by a narrowband converter using gate arrays. The work is divided into two sections, the first one dealing with the theoretical background. The second part denotes the realization of the design. The result of this work is a signal analyzer in a FPGA controlled from a computer application via the USB 3.0 interface.

KEYWORDS

FPGA, USB, AD converter, FFT, Spectrum, FIR filter, CIC filter, Radio, DDC, Demodulation, RAM Memory, DDR3

KRAUS, V. *Analyzátor signálu s FPGA*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 64 s. Vedoucí diplomové práce doc. Ing. Pavel Šteffan, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Analyzátor signálu se záznamem jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce doc. Ing. Pavlu Šteffanovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Dále bych chtěl poděkovat odbornému konzultantovi Ing. Michalu Kováčovi Ph.D. za odborné vedení, konzultace a užitečné rady k práci.

V Brně dne

.....

(podpis autora)

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře
vybudované v rámci projektu CZ.1.05/2.1.00/03.0072
Centrum senzorických, informačních a komunikačních systémů (SIX)
operačního programu Výzkum a vývoj pro inovace.

OBSAH

Úvod	10
1 Teoretická část	11
1.1 Programovatelná hradlová pole	11
1.2 Fourierova transformace	12
1.3 Číslicová filtrace signálu	13
1.3.1 Filtr Klouzavý průměr	14
1.3.2 CIC filtry	15
1.3.3 FIR filtr	16
1.4 Směšování	18
1.5 Syntezátor	19
1.5.1 DDS – Direct digital synthesizer	19
1.6 Digital down converter	20
1.7 Modulace signálu	20
1.7.1 Amplitudová modulace AM	20
1.7.2 Frekvenční modulace FM	21
1.8 Vzorkovací teorém a aliasing	22
1.9 Paměť typu DDR3 SDRAM	23
1.10 Univerzální sériová sběrnice	24
2 Praktická část	26
2.1 Volba hardware	26
2.1.1 Použitý vývojový kit	26
2.1.2 FPGA modul Mercury KX1	27
2.1.3 AD převodník AD9680-500	28
2.2 Návrh systému	30
2.2.1 Režimy	30
2.2.2 Řízení systému	31
2.2.3 Řídící registry	32
2.2.4 Adresní prostor	34
2.2.5 Komunikační blok	34
2.3 Komunikace po USB 3.0	34

2.3.1	USB 3.0 Cypress FX3.....	34
2.3.2	Komunikační protokol	36
2.3.3	Formáty odesílaných dat	37
2.3.4	USB blok.....	39
2.4	Bloky jednotlivých režimů.....	41
2.4.1	Blok pro režim Spectrum	41
2.4.2	Blok pro Úzkopásmový režim	42
2.4.3	Blok pro Sample režim	48
2.5	Oživení zařízení	50
2.5.1	USB komunikace	50
2.5.2	Úzkopásmový režim	51
2.5.3	RAM paměť	53
2.5.4	Spektrum	54
2.6	Navržená aplikace	54
2.6.1	Záložka Spectrum	54
2.6.2	Záložka Radio	56
2.6.3	Záložka Sampling	56
3	Závěr	58
	Literatura	59
	Seznam obrázků	62
	Seznam tabulek	64

ÚVOD

V dnešní době jsou digitální technologie již prakticky rozšířeny do všech odvětví lidské činnosti. Jejich využití v současných zařízeních je způsobeno mnoha výhodami oproti analogovému řešení. Konkrétně menší rozměry, nižší hmotnost, vyšší spolehlivost, možnost přizpůsobení parametrů na úrovni software bez hardwarových úprav. Vývoj digitálních technologií v dnešní době nabízí stále výkonnější digitální součástky, které lze použít při zpracování signálu v reálném čase. Z tohoto důvodu se i v oblasti radiové techniky odchází od analogového řešení a přechází se na digitální systémy. Možností je použití signálových procesorů nebo hradlových polí FPGA.

Cílem tohoto projektu je navrhnout a implementovat spektrální analyzátor rozšířený o záznam signálu do DDR paměti za využití hradlového pole. Práce je dělena na dvě části. První část je teoretická a zabývá se tématy potřebnými pro daný projekt. Představuje paměti SDRAM a srovnání jejich typů. Dále komunikaci po USB 3.0 ve srovnání se staršími standardy. Popisuje zpracování signálů v hradlovém poli, jako jsou filtry, Fourierova transformace pro získání frekvenčního spektra, digital down converter, atd. Důležitý je také Antialiasing a dodržení vzorkovacího teorému, aby došlo ke správné interpretaci přijímaného signálu.

Ve druhé části práce se přistupuje k návrhu projektu a řešení jednotlivých dílčích bloků. Cílem pro tento projekt byly tři funkční režimy. Vzorkovací sample režim pro záznam signálu, přehled frekvenčního spektra nebo úzkopásmový přijímač. Praktická část obsahuje popis zvoleného hardwaru, bloková schémata, stavové automaty nebo popsany komunikační protokol. Poslední částí práce je realizace počítačové aplikace, která celý systém řídí a zajišťuje zobrazení zpracovaných dat.

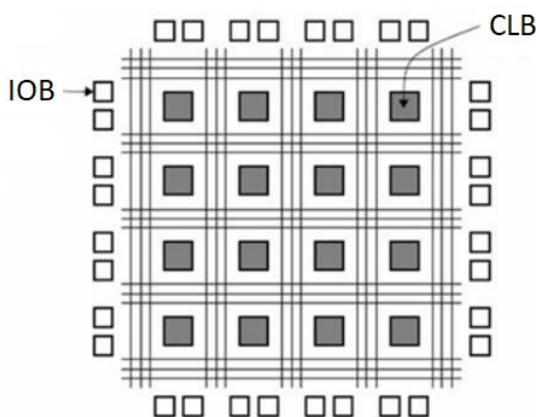
1 TEORETICKÁ ČÁST

První část této práce je teoretická. Seznamuje s technologiemi vhodnými pro implementaci spektrálního analyzátoru dle zadání. Nejdříve popisuje technologii hradlových polí a poté paměť typu DDR3. Dále následuje zpracování signálů jako je například filtrace, Fourierova transformace, nebo směšování. Poslední částí teoretického rozboru je popis univerzální sériové sběrnice neboli USB.

1.1 Programovatelná hradlová pole

Programovatelné hradlové pole neboli FPGA (Field Programmable Gate Array) je typem digitálního integrovaného obvodu. Spadá do programovatelných logických obvodů a obsahuje logické bloky propojené konfigurovatelnou maticí spojů. Ta umožňuje bloky navzájem propojit a vytvořit libovolné číslicové zařízení (procesor, řídicí obvod osciloskopu). Hlavní výhodou oproti mikrokontrolérům je jejich pracovní kmitočet. Nevýhodou naopak vyšší cena.

FPGA se dnes stávají oblíbené pro svoji flexibilitu a programovatelnost. Hlavními jazyky pro jejich programování jsou VHDL nebo Verilog. Typicky se hodí do malosériové výroby, kde se nevyplatí ASIC obvody a řešení s procesorem není vhodné. Další možnost využití je například prototypování složitějších obvodů. [1], [14], [15]

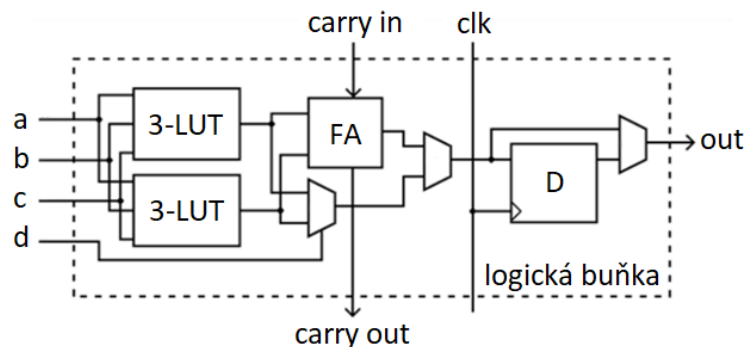


Obr. 1: Struktura hradlového pole FPGA [14]

Základními stavebními prvky obvodů FPGA jsou logické buňky (CLB), I/O obvody (IOB) a propojovací kanály.

Logické buňky CLB jsou nejpodstatnější částí FPGA, slouží totiž pro vytváření jednotlivých logických funkcí. Obvykle se skládají z vyhledávací tabulky LUT (Lookup table), úplné sčítačky a klopného obvodu typu D. [1], [2], [15]

Vyhledávací tabulka je tvořena pamětí typu SRAM a obvykle má 4 nebo 6 vstupů. Plní základní úlohu programovatelných obvodů, protože právě LUT v sobě uchovává pravdivostní tabulku vytvářených funkcí.



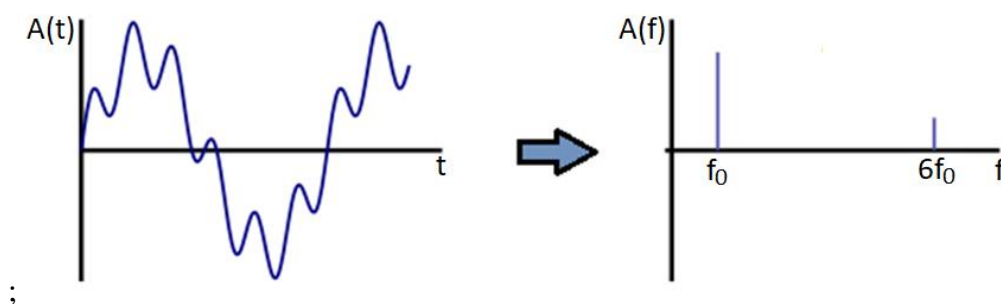
Obr. 2: Struktura logické buňky hradlového pole [2]

Počet logických buněk určuje složitost FPGA obvodu a tím i jeho cenu. Výrobce nabízí vždy několik variant a při volbě FPGA je vhodné se dívat na tento počet. Jejich nevyužití zvyšuje cenu obvodu a také neefektivní implementaci.

Dále FPGA obsahují i další pomocné obvody, jako vysokorychlostní transceivery nebo třeba DSP (Digital Signal Processing). Využití DSP je užitečné při různých matematických operacích. Využívá se pro modulátory, filtry, násobičky, transformace nebo DDS (Direct Digital Synthesis), kdy tímto nedochází k velké spotřebě LUT. Výhodou také je vyšší rychlost zpracování, než při využití implementace pomocí vyhledávacích tabulek. Ty dražší FPGA mohou obsahovat například i zabudovaný mikroprocesor. [1], [2], [14], [15]

1.2 Fourierova transformace

Jedná se o matematickou metodu, která slouží pro převod posloupnosti vzorků signálu z časové oblasti do frekvenční oblasti, obecně známé jako spektrum signálu. Tento převod je ve zpracování signálů velice důležitý, ve frekvenční oblasti se nabízí mnohem více možností na úpravu signálu. [4], [18], [19]



Obr. 3: Příklad zobrazení spektra signálu složeného ze dvou sinusových průběhů [4]

Diskrétní Fourierova transformace (DFT) pracuje s digitálním signálem. Vstupními hodnotami pro Fourierovu transformaci jsou diskrétní hodnoty vzorkovaného spojitého signálu. Výstupem jsou poté jednotlivé frekvence obsažené ve spektru analyzovaného signálu a jejich amplitudy. Nevýhodou DFT je velká časová náročnost, vyžaduje totiž N^2 součinů a N^2 součtů. S rostoucím počtem vzorků prudce narůstá i počet operací a toto

množství výrazně snížilo možnost aplikace v reálném čase. Následující rovnice obsahuje předpis pro DFT. [4], [18], [19]

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j \cdot 2\pi \cdot \frac{k}{N} n} \quad (1)$$

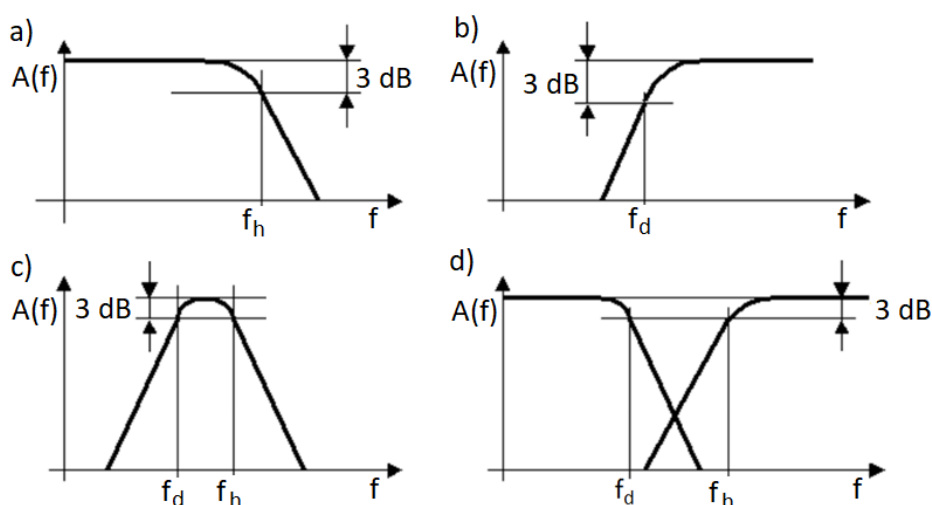
X_k	... k-tá harmonická frekvence ve frekvenční oblasti
x_n	... n-tý vzorek signálu v časové oblasti
k	... pořadí výstupního vzorku
n	... pořadí vstupního vzorku
N	... velikost vektoru vstupních hodnot

Změna nastala v roce 1965, kdy James Cooley a John Tukey popsali nový a velmi efektivní algoritmus pro výpočet DFT. Ten výrazně snížil počet matematických operací a dostal název Rychlá Fourierova transformace neboli FFT. Ta se brzy uplatnila v mnoha odvětvích, kde je potřeba pracovat se signálem. Díky úspoře výpočetních kroků je vhodná jak do signálových procesorů, tak pro přímou implementaci na čip. [4], [18], [19]

1.3 Číslicová filtrace signálu

Filtrace se používá ke dvěma hlavním účelům. K oddělení několika sloučených signálů nebo k obnově zkresleného signálu. Oddělení signálů se používá, když je signál ovlivněn šumem, jinými signály nebo došlo k nějaké interferenci. Číslicové filtry se dělí na dva hlavní typy. Filtry s konečnou impulzní odezvou FIR a s nekonečnou impulzní odezvou IIR. Výhoda IIR filtrů je v nižší výpočetní složitosti oproti FIR filtrům, ovšem u nich není zaručena stabilita.

Podle amplitudově frekvenčních charakteristik rozlišujeme filtry na 4 základní druhy. [6], [10],



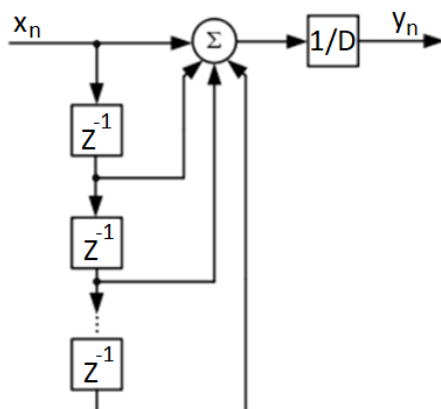
Obr. 4: Filtry typu: a) dolní propust, b) horní propust, c) pásm. propust, d) pásm. zádrž [6]

1.3.1 Filtr Klouzavý průměr

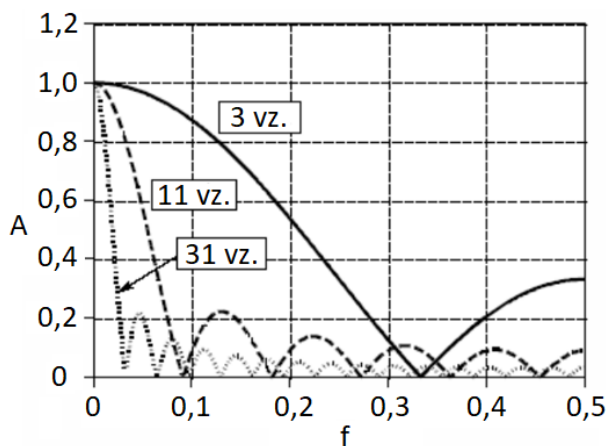
Filtr typu klouzavý průměr využívá průměrování. Provádí se z daného vzorku a N vzorků předchozích. Zachovává nižší frekvence a potlačuje vyšší. Slouží pro potlačení periodického rušení superponovaném na konstantním signálu nebo k potlačení krátkodobých výkyvů. Jeho výhodou je jednoduchost. Lze ho použít opakovaně, tzn. řadit filtry kaskádně. S každým vyšším stupněm dochází k lepšímu odstranění šumu, nevýhodou ovšem je, že rozmazává ostré přechody. [10], [11]

Filtr typu klouzavý průměr lze popsat vztahem:

$$y_n = \frac{1}{N} (x_n + x_{n-1} + \dots + x_{n-N+1}) = \frac{1}{N} \sum_{n=0}^{N-1} x_n \quad (2)$$



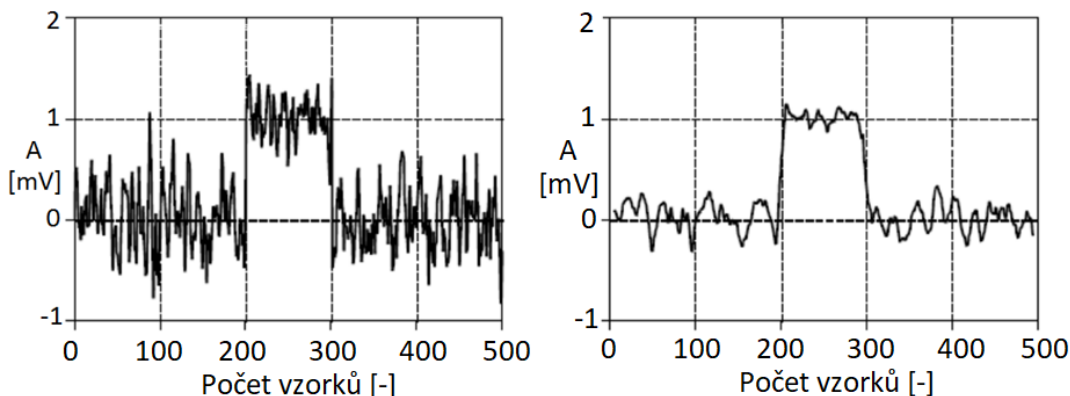
Obr. 5: Struktura filtru s klouzavým průměrem [10]



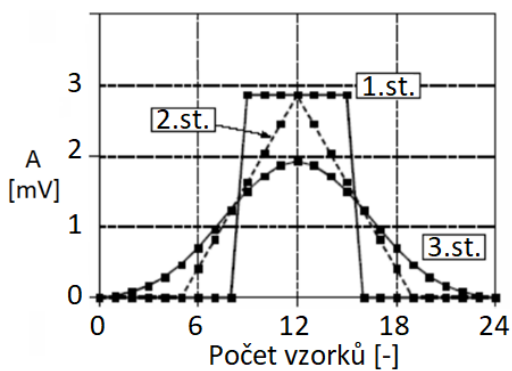
Obr. 6: Frekvenční charakteristika filtru [11]

Zmíněnému filtru se přesně říká Jednoduchý, existují i další varianty:

- Exponenciální klouzavý průměr
- Vyhlazený klouzavý průměr
- Vážený klouzavý průměr



Obr. 7: Průměrování 12ti vzorků [11]

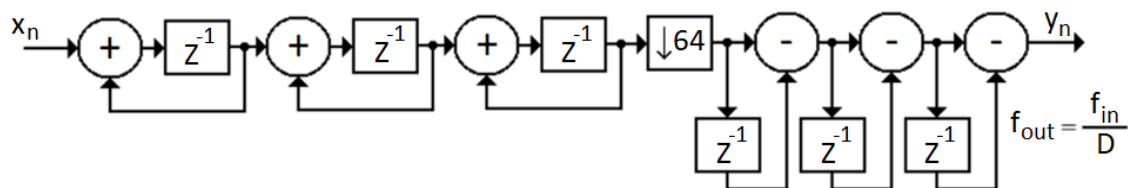


Obr. 8: Výsledky průměrování při kaskádním řazení filtrů [11]

Na předchozích obrázcích je vidět filtrace za použití filtru Klouzavý průměr. Je patrné, že šum odstraňuje poměrně dobře, při použití vícebodového průměru by šum vymizel téměř úplně. Značnou nevýhodou je ovšem zkosení hran pulzu, pokud by pulz byl hodně krátký, mohl by vymizet úplně. Dále je také vidět kaskádní řazení filtrů. Se stoupajícím stupněm dochází k silnému zkosení ostrých přechodů. [10], [11]

1.3.2 CIC filtry

Jinak řečeno kaskádově integrovaný hřebenový filtr. Tento druh filtru se využívá při implementaci decimátoru, slouží pro snížení vzorkovací frekvence. Jedná se o filtr s konečnou impulzní charakteristikou. Jeho struktura je oproti FIR filtru poměrně jednoduchá, nepotřebuje uchovávat koeficienty ani neobsahuje žádné násobičky signálu. [9], [10], [15],



Obr. 9: Struktura filtru CIC třetího řádu s decimačním faktorem 64 [9]

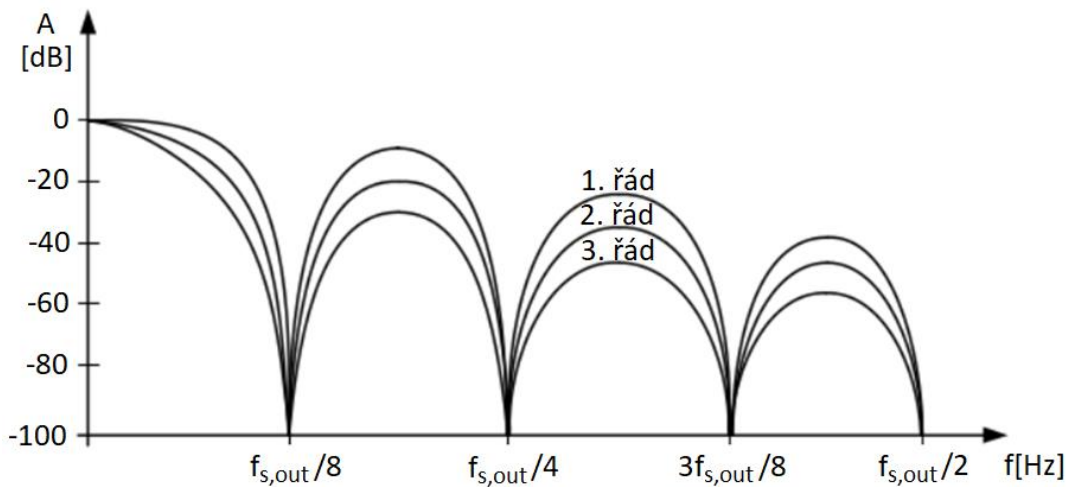
Vztah popisující CIC filtr lze zapsat pomocí vzorce:

$$y_n = (x_n - x_{n-D}) + y_{n-1} \quad (3)$$

Přenosová funkce je dána vztahem:

$$H(z) = \frac{1-z^{-D}}{1-z^{-1}} \quad (4)$$

D ... decimální faktor



Obr. 10: Frekvenční odezva CIC filtru [10]

Nejčastěji se CIC filtr využívá v kaskádní kombinaci s FIR filtrem, což přináší výhody oproti použití samostatných filtrů. CIC filtr dokáže zpracovat signály s vysokými vzorkovacími kmitočty a ty snížit. FIR filtr zase lépe pracuje na nižších vzorkovacích kmitočtech, má zpravidla strmější charakteristiky a implementace v FPGA není tak náročná na počet využitých bloků. CIC filtry mají ovšem velký zisk, proto jsou ještě před FIR filtrem bloky pro konverzi na příslušnou bitovou šířku. [9], [10], [15]

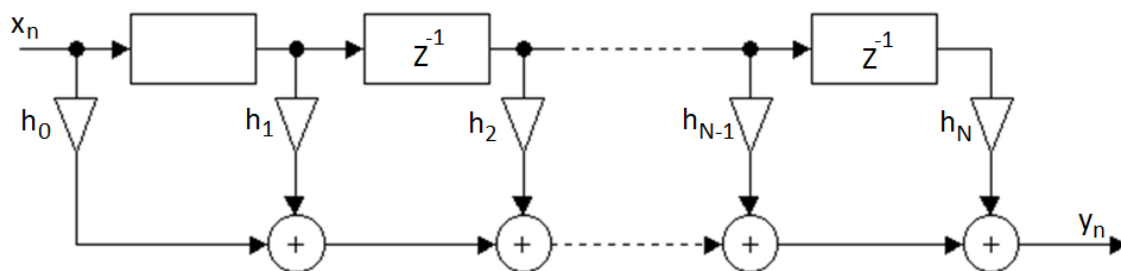
1.3.3 FIR filtr

Jedná se o lineární diskrétní filtry s konečnou impulzní odezvou (Finite Impulse Response). Jsou to plně číslicové filtry, které nemají ekvivalent v analogové oblasti zpracování signálů. Vstupem pro FIR filtr jsou diskrétní vzorkovaná data z AD převodníku. Typy mohou být dolní, horní, pásmová propust nebo zádrž. Obvykle se realizují nerekurzivním způsobem, tzn. neobsahují žádnou zpětnou vazbu, tudíž FIR jsou absolutně stabilní. Jejich výstup je definován jako konvoluční suma a popisuje ji následující vztah. [8], [15], [20]

$$y_n = h_0 \cdot x_n + h_1 \cdot x_{n-1} + \dots + h_{N-1} \cdot x_{n-(N-1)} = \sum_{i=0}^{N-1} h_i \cdot x_{n-i} \quad (5)$$

y_n ... výstup (odezva) lineárního filtru

h_i ... hodnoty impulzní charakteristiky, systémové koeficienty



Obr. 11: Struktura FIR filtru [8]

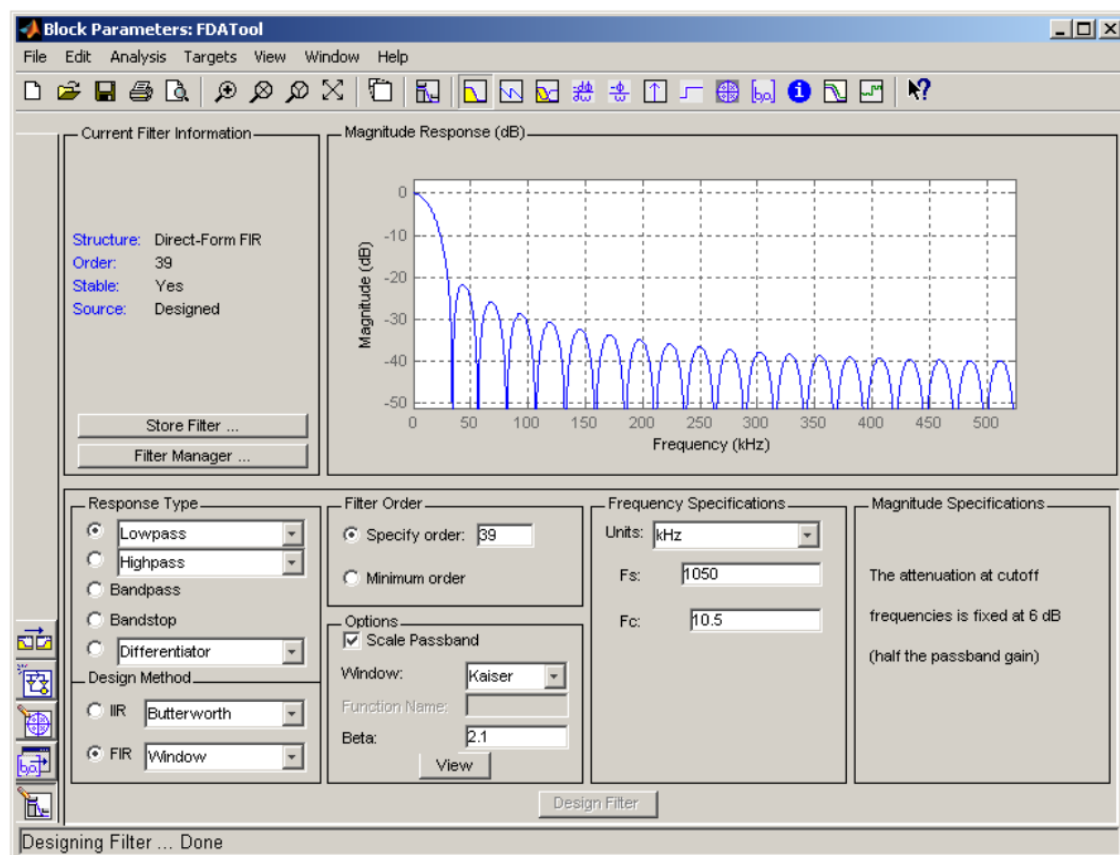
Obraz diferenční rovnice:

$$Y(z) = \sum_{n=0}^{N-1} h_n \cdot X(z) \cdot z^{-n} \quad (6)$$

Přenosová funkce FIR filtru:

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{n=0}^{N-1} h_n \cdot z^{-n} \quad (7)$$

Nejjednodušší možností návrhu koeficientů pro FIR filtr je využití prostředí Matlab a nástroje Fdatool. Ten slouží pro návrh různých číslicových filtrů. Náhled nástroje je možné vidět na následujícím obrázku.



Obr. 12: Ukázka prostředí Fdatool při návrhu FIR filtru [21]

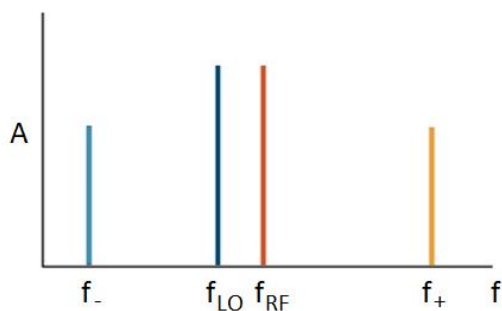
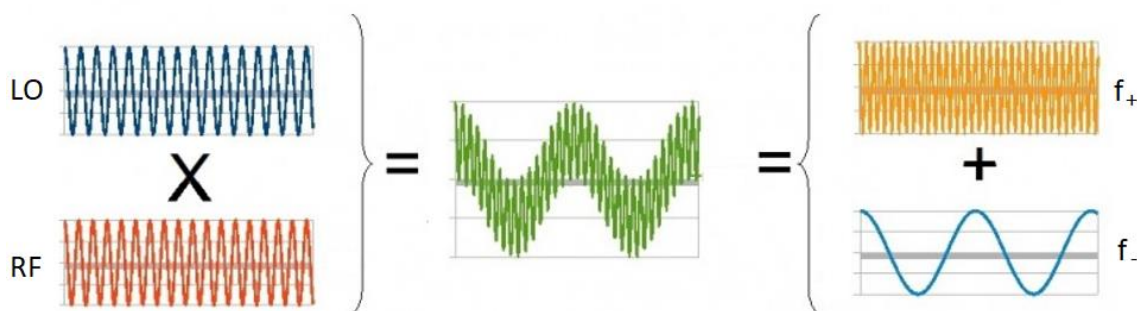
1.4 Směšování

Směšování signálů je důležitou součástí při signálovém zpracování. Slouží pro převedení signálu na určitém kmitočtu do jiného kmitočtového pásma, tzv. na mezifrekvenci, kde se se signálem lépe pracuje, např. zesilování nebo filtrace.

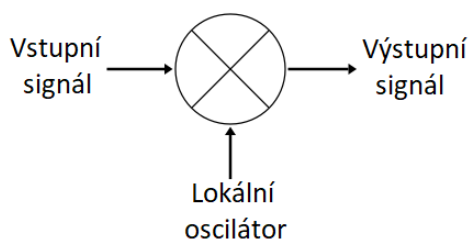
Směšovač může být jak analogový, tak i digitální. Principem je směšovat přijímaný signál s lokálním oscilátorem o určitém kmitočtu. Tímto vznikají dva nové signály. Jedním je signál s rozdílovou frekvencí a druhý se součtovou. V digitálních obvodech se směšování zajišťuje pomocí dvouvstupové násobičky. [34], [37], [38]

$$f_- = f_{RF} - f_{LO} \quad (8)$$

$$f_+ = f_{RF} + f_{LO} \quad (9)$$



Obr. 13: Princip směšování [34]



Obr. 14: Značka pro směšování

1.5 Syntezátor

Frekvenční syntezátory slouží pro generování harmonických signálů různých kmitočtů, nejčastěji sinusového nebo pravoúhlého průběhu. Využívají se jako lokální oscilátory při směšování. Jejich výhodou oproti klasickým oscilátorům je možnost přeladění, kdy kmitočet generovaného signálu lze měnit jen s určitým krokem. V dnešní době jsou syntezátory vyráběny ve formě integrovaných obvodů a jejich frekvenční rozsah je v řádu GHz. Nejčastěji se používají syntezátory s fázovým závěsem neboli PLL, které jsou populární pro svoji nízkou cenu. Využívají technologii smyčky fázového závěsu, obsahují napětím řízené oscilátory, děliče kmitočtu, filtry, fázové detektory nebo další bloky. Druhou variantou jsou přímé číslicové syntezátory DDS, jejichž výhodou oproti PLL je rychlost přeladění. Obvody FPGA umožňují implementaci syntezátorů tohoto typu. [43]

1.5.1 DDS – Direct digital synthesizer

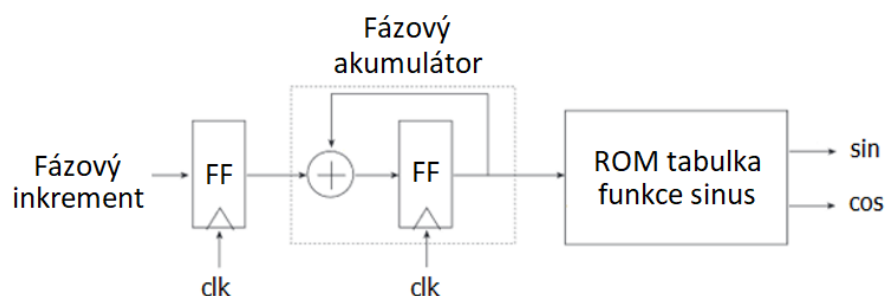
Jedná se o typ kmitočtového syntezátoru, který pracuje na principu přímé kmitočtové syntézy. Signál je zde generován na základě vyčítaných hodnot uložených v paměti ROM. Důležitým prvkem je fázový akumulátor, který se inkrementuje s každým hodinovým taktem. Na základě této hodnoty dochází k vyčítání diskretních hodnot z paměti ROM. Ta obsahuje uložený průběh funkce sinus (cosinus). Při použití v analogových obvodech je syntezátor rozšířen D/A převodníkem, který převádí diskretní signál na spojitý, jeho nevýhodou je ovšem schodovitost. Proto je použit filtr pro vyhlazení signálu. Vstupní hodnotou těchto syntezátorů bývá tzv. fázový inkrement $\Delta\theta$. Rozlišení kmitočtu syntezátoru závisí na kmitočtu hodinového signálu f_{clk} a na bitové šířce fázového inkrementu B. [36], [39], [40]

$$\Delta f = \frac{f_{clk}}{2^B} \quad (10)$$

Výstupní kmitočet je poté dán vztahem:

$$f_{out} = \Delta f \cdot \Delta\theta = \frac{f_{clk} \cdot \Delta\theta}{2^B} \quad (11)$$

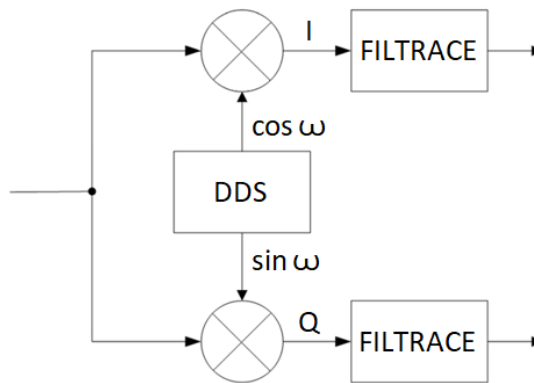
Pokud by se vstupní hodnotou zvolilo $\Delta\theta = 1$, došlo by k vygenerování minimálního možného kmitočtu, který je roven rozlišení Δf .



Obr. 15: Architektura syntezátoru DDS

1.6 Digital down converter

Digital down converter neboli DDC se využívá v digitálním zpracování signálu. Cílem je převést digitalizovaný signál na nižší frekvenci o nižším vzorkovacím kmitočtu. Základním blokem je IQ demodulátor, který rozdělí přijímaný signál na reálnou a imaginární složku. Pomocí směšování přijímaného signálu s harmonickými signály $\sin \omega$ a $\cos \omega$, odpovídajícím kmitočtu přijímaného signálu, dostaneme signál do základního pásma. Poté následují filtry pro snížení vzorkovacího kmitočtu a odfiltrování zbylých harmonických složek. Běžně se používá kombinace decimačního filtru CIC a FIR filtru typu dolní propust. [22], [44]



Obr. 16: Základní struktura DDC

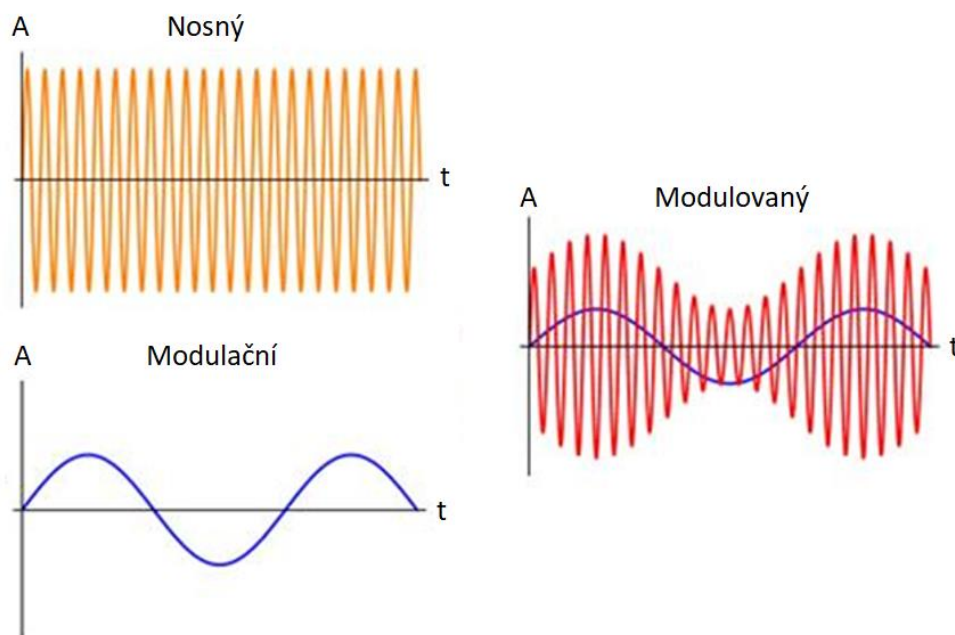
Obvykle se za blokem DDC nachází obvody pro demodulaci signálu. Může se jednat například o AM nebo FM demodulátory.

1.7 Modulace signálu

Jedná se o proces, při kterém dochází ke změně některé z vlastností nosného signálu pomocí modulačního signálu (frekvence, amplituda, fáze). Toto se využívá zejména u radiového přenosu. Cílem je na nosný signál namodulovat přenášené informace. Běžně se s modulací setkáme například u mobilního telefonu, televizního nebo radiového vysílání. [35], [41], [42]

1.7.1 Amplitudová modulace AM

Historicky se jedná o nejstarší druh modulace. V závislosti na změně modulačního signálu dochází ke změně amplitudy nosného signálu. Při této modulaci nedochází k ovlivňování frekvence ani fáze nosného signálu. Lze říci, že obálka modulovaného signálu odpovídá tvarem signálu modulačnímu. Toto lze vidět na obrázku na následující straně. Kmitočet nosného signálu je vždy podstatně vyšší než kmitočet modulačního. V dnešní době se již AM vysílání moc nevyužívá z důvodu náchylnosti na rušení. [35], [42]



Obr. 17: Princip amplitudové modulace [42]

Matematický popis amplitudové modulace lze vyjádřit vztahem:

$$y_t = (1 + M \cdot \cos(\omega_m t + \varphi)) \cdot N \cdot \sin(\omega_n t) \quad (12)$$

M ... amplituda modulačního signálu

N ... amplituda nosného signálu

ω_M ... kmitočet modulačního signálu

ω_N ... kmitočet nosného signálu

φ ... fázový posuv vůči nosnému signálu

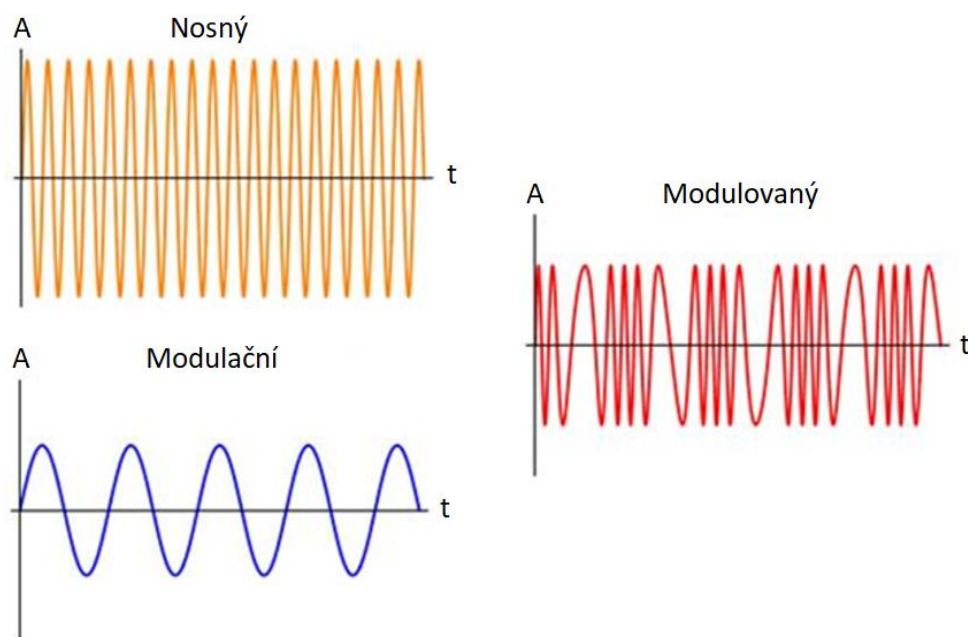
Měřítkem působení modulačního signálu na nosnou vlnu je tzv. hloubka modulace:

$$m = \frac{M}{N} \quad (13)$$

Často se udává v procentech.

1.7.2 Frekvenční modulace FM

Při frekvenční modulaci dochází ke změně hodnoty frekvence nosného signálu na základě amplitudy modulačního signálu. Fáze ani amplituda nosného signálu nejsou změněny. Maximální amplitudě modulačního signálu odpovídá maximální změna frekvence nosného signálu. Tato změna se nazývá kmitočtový zdvih Δf . Frekvenční modulace se používá zejména pro radiový rozhlas. [35], [41]



Obr. 18: Princip frekvenční modulace [41]

Frekvenční zdvih:

$$\Delta f = k \cdot M \quad (14)$$

Měřítkem působení modulačního signálu na nosnou vlnu je tzv. modulační index:

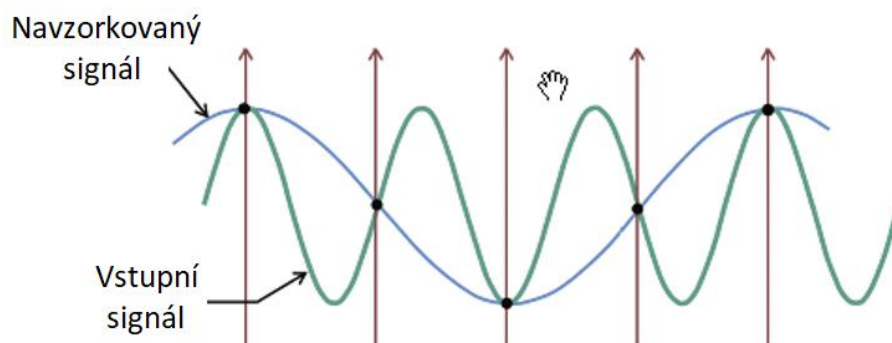
$$h = \frac{\Delta f}{f_m} \quad (15)$$

Matematický popis frekvenční modulace lze vyjádřit vztahem:

$$y_t = N \cdot \cos(\omega_n t + M \cdot h \cdot \sin(\omega_m t)) \quad (16)$$

1.8 Vzorkovací teorém a aliasing

Nyquistův vzorkovací teorém říká, že rekonstrukce signálu je možná pouze v případě, když je vzorkovací kmitočet dvakrát vyšší než maximální frekvence vzorkovaného signálu. Při použití nižšího vzorkovacího kmitočtu neboli podvzorkování by se v důsledku tzv. aliasingu mohly objevit ve spektru nežádoucí signály. Při převzorkování musí být vzorkovací kmitočet dvakrát větší, než je šířka pásma převzorkovaného signálu.



Obr. 19: Podvzorkování signálu [45]

Při nízké vzorkovací frekvenci jsou vysoké kmitočty v původním signálu špatně interpretovány. Vysokofrekvenční signály se ve spektru objeví jako nízké a budou od nízkofrekvenčních nerozpoznatelné, což vede ke zkreslení a znehodnocení vzorkovaných dat. Z tohoto důvodu se používají tzv. antialiasingové filtry typu dolní propust, které propouštějí pouze kmitočty menší, než je polovina vzorkovací frekvence. Jedná se o analogové filtry ještě před AD převodníkem. [23], [24]

1.9 Paměť typu DDR3 SDRAM

Daný typ paměti spadá do kategorie SDRAM (Synchronous Dynamic Random-Access Memory). Jedná se o paměti synchronní, což znamená, že potřebují externí řídicí hodinový signál, podle kterého probíhá přenos dat. Prvním typem této paměti byla paměť SDR (Single Data Rate), ovšem ji brzy nahradil nástupce s názvem DDR (Dual Data Rate). Z názvu již vyplývá, že se jedná o dvojitý datový zápis. Ten probíhá jak na sestupnou, tak na náběžnou hranu hodinového signálu. V praxi se můžeme setkat s několika generacemi DDR pamětí. Nejnovějšími modely jsou DDR5. Lze ale také najít moduly GDDR, z anglického slova Graphics, speciálně navrženy pro grafické karty. [3], [5], [16], [17]

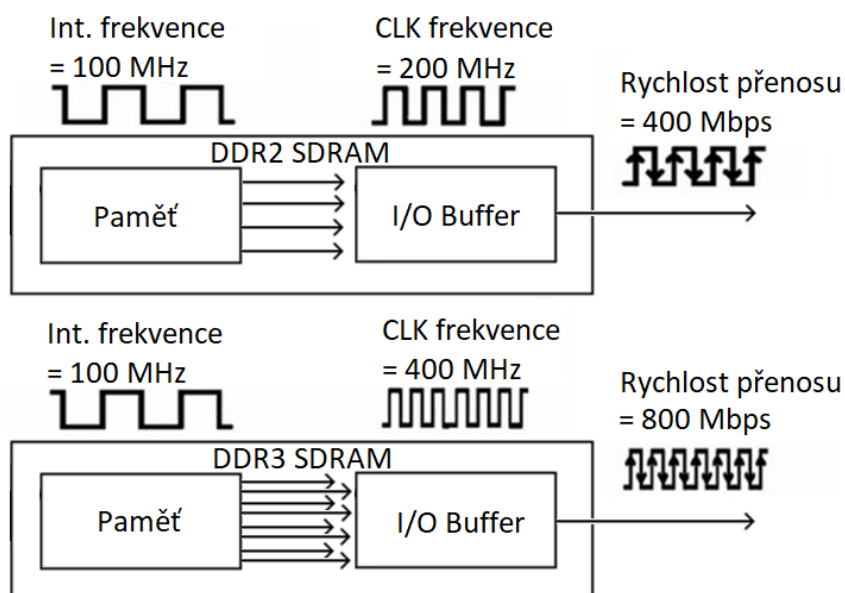
Tab. 1: Vývoj RAM pamětí [3]

Standard	CLK interní [MHz]	CLK sběrnice [MHz]	Rychlost přenosu [MT/s]	Propustnost [GB/s]	Napětí [V]
SDRAM	100-166	100-166	100-166	0,8 - 1,3	3,3
DDR	133-200	133-200	266-400	2,1 - 3,2	2,5/2,6
DDR2	133-200	266-400	533-800	4,2 - 6,4	1,8
DDR3	133-200	533-800	1066-1600	8,5 - 14,9	1,35/1,5
DDR4	133-200	1066-1600	2133-3200	17 - 21,3	1,2

Jak vyplývá z tabulky na předchozí straně. Cílem každé další generace je zvyšovat pracovní frekvenci a tím i přenosovou rychlost. Napájecí napětí se naopak snižuje.

Rychlost DDR pamětí je dána frekvencí hodinového signálu. Jak bylo řečeno výše, moduly využívají jak sestupné, tak náběžné hrany, čímž je rychlost přenosu dvojnásobná

oproti hodinovému signálu. Pro DDR3 je rychlost sběrnice až 800 MHz, z čehož plyne rychlost přenosu až 1600 MT/s (megatransferů za vteřinu). Při náhledu do tabulky si lze všimnout interní frekvence, která je pouze do 200 MHz, zatímco komunikace s DDR3 probíhá na frekvenci 800 MHz. Je to způsobeno vstupními a výstupními buffery, která data paralelizují/serializují, jak je vidět na následujícím obrázku. [3], [5], [16], [17]



Obr. 20: Zpracování dat pomocí DDR2 a DDR3 [5]

1.10 Univerzální sériová sběrnice

Zkratka USB pochází ze slova Universal Serial Bus. Jedná se o univerzální sériovou sběrnici, která slouží pro připojení periférií k počítači (tiskárny, myši, klávesnice, fotoaparátu atd.). Standardně mělo USB 4 vodiče, DATA+, DATA-, VCC, GND a kabel byl stíněn hliníkovou folií. Nyní má USB několik verzí, liší se jak v počtu vodičů, tak v přenosových rychlostech i přeneseném výkonu. V následující tabulce jsou vidět jednotlivé typy USB. [12], [25], [26]

Tab. 2: Přehled verzí USB [25]

Verze	Název	Rok vydání	Přenosová rychlost	Konektor	Počet kontaktů
USB 1.0	Low Speed	1996	1,5Mbit/s	Typ-A	4
USB 1.1	Full Speed	1998	12Mbit/s	Typ-A	4
USB 2.0	High Speed	2000	480 Mbit/s	Typ-A	4
USB 3.0 (3.1 Gen 1)	SuperSpeed	2008	5 Gbit/s	Typ-A SuperSpeed	9
USB 3.1 (3.1 Gen 2)	SuperSpeed+	2013	10 Gbit/s	Typ-C	24
USB 3.2	SuperSpeed+	2017	20 Gbit/s	Typ-C	24

Typy konektorů výše v tabulce jsou pouze základní, setkat se dá i s dalšími variantami. Příkladem je Typ-B, nebo různé velikostní varianty (Micro, Mini). Počet kontaktů je v tabulce uveden přímo na odpovídající konektor a u jiných se může lišit. Z důvodu zvyšování přenosové rychlosti je u vyšších verzí USB použito více kontaktů (vodičů). U USB 3.1 nebo 3.2 však nemusí být využity všechny. Závisí na koncovce na druhé straně kabelu. Cílem u vyšších verzí bylo také zvýšit přenášený výkon až na 100 W, zatímco USB 3.0 má pouze 15 W. Došlo by tak ke zvýšení rychlosti nabíjení externích zařízení. [12], [25], [26]



Obr. 21: Vybrané typy USB konektorů [12]

Nejpoužívanější jsou v dnešní době verze 2.0 a 3.0, které jsou spolu kompatibilní. Na první pohled konektory vypadají stejně. Rozdíl je v barvě, kdy USB 3.0 má modrou barvu výplně a obsahuje více kontaktů.

I když je USB hojně rozšířeno a specifikace její komunikace je dostupná, implementace je poměrně složitá. Dnes existují tzv. FTDI převodníky, které převádí komunikaci USB na nějakou vhodnější. Příkladem pro verzi 2.0 může být převod na sériovou linku RS-232, pro niž existuje řada knihoven a není problém tedy komunikovat po USB 2.0. U verze 3.0 je to složitější, existují FTDI s výstupem přes FIFO, u kterých je vhodné vyčítání dat pomocí rychlého FPGA. Nebo existují speciální programovatelné moduly a lze je vhodně upravit pro požadovanou aplikaci. [12], [25], [26]

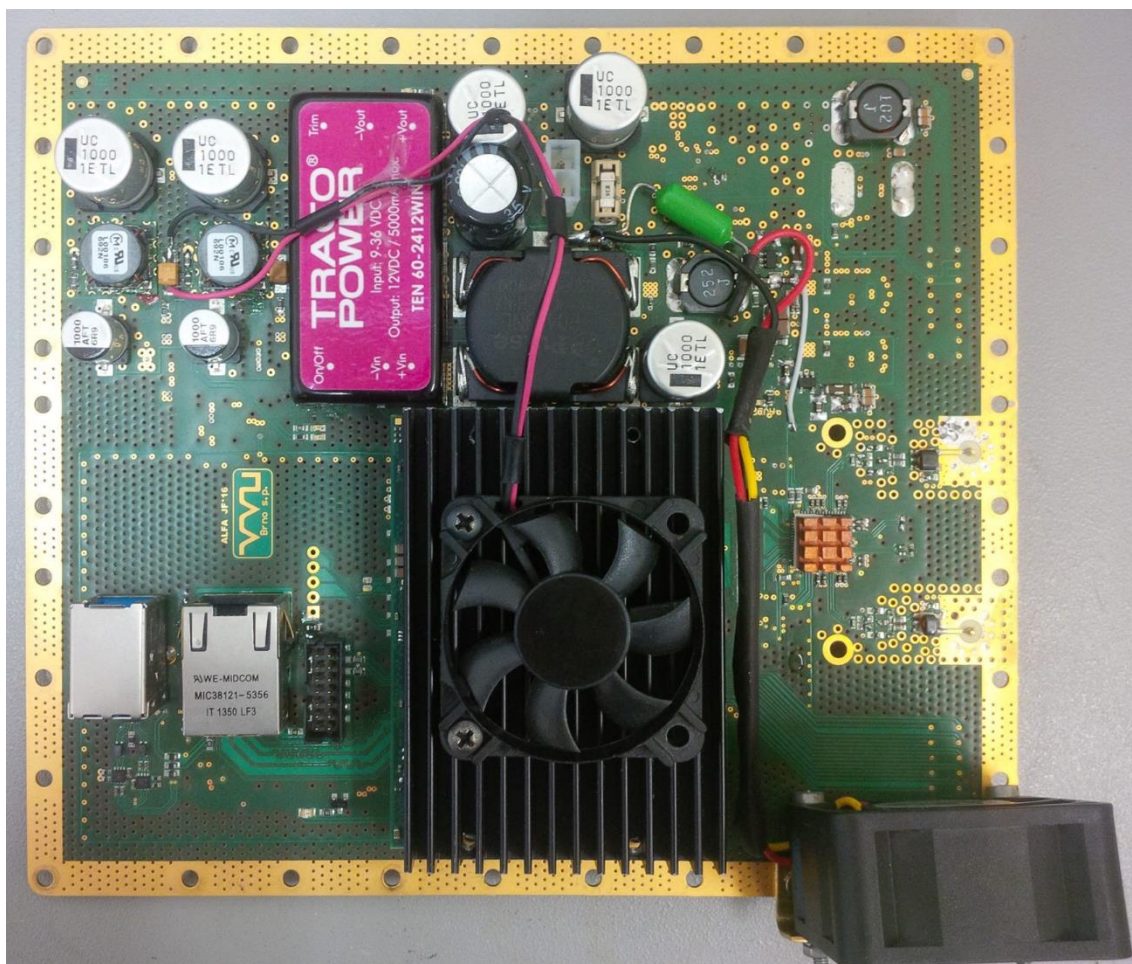
2 PRAKTICKÁ ČÁST

2.1 Volba hardware

Pro výběr modulu s hradlovým polem jsou rozhodující různé parametry, rychlost, kapacita obvodu, dostupná komunikační rozhraní, RAM paměť, počet vstupně/výstupních pinů. Podle zadání bylo důležité, aby modul obsahoval dostatečně velkou RAM paměť, rozhraní USB 3.0 a FPGA z hlediska počtu logických buněk vyhovovalo velkému početnímu výkonu (FFT, filtry, Cordic jádra).

2.1.1 Použitý vývojový kit

Pro realizaci této práce byla zvolena vývojová deska doporučená vedoucím, která je vidět na následujícím obrázku.



Obr. 22: Zvolený vývojový modul

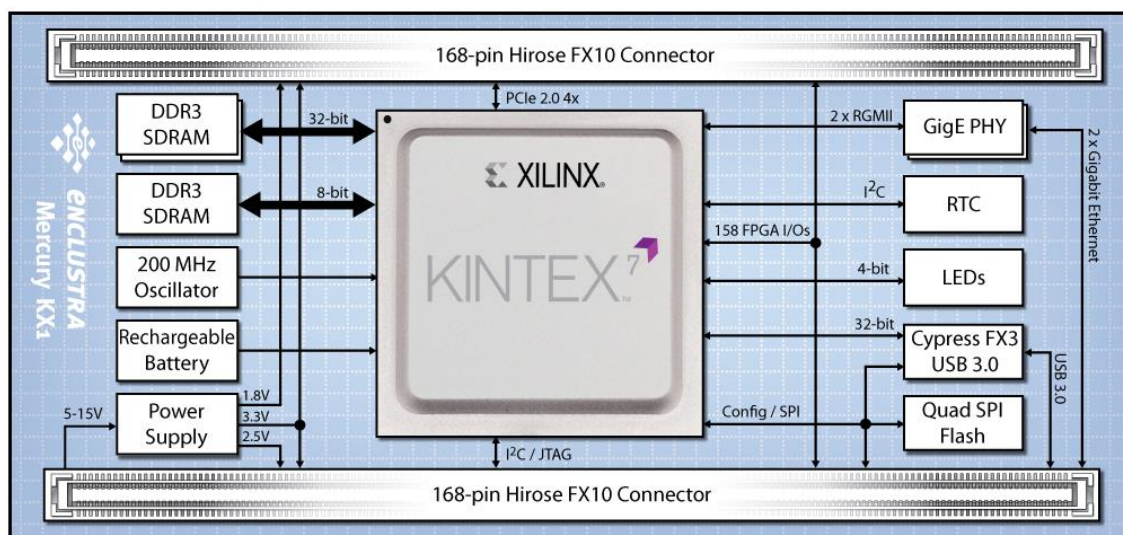
Napájení desky je 9-36 V. Pro komunikaci jsou zde vyvedeny konektory USB 3.0 a RJ-45. Ten umožňuje komunikaci po ethernetu 1 Gbit/s. Dále obsahuje dvoukanálový

převodník AD9680 se vzorkovací rychlostí 500 MS/s. Vývojový kit také obsahuje konektory pro připojení druhé desky osazené syntezátorem, směšovači a zesilovači pro rozšíření frekvenčního rozsahu. Druhá deska ovšem není cílem této práce a její využití je plánováno do budoucna.

Jak bylo řečeno výše, důležité je vhodně zvolit FPGA. Vývojový kit umožňuje osazení modulem Mercury KX1 od firmy Enclustra, který obsahuje FPGA čip třídy Kintex-7.

2.1.2 FPGA modul Mercury KX1

Modul Mercury KX1 je osazen čipy FPGA Kintex-7 a na výběr je z několika variant FPGA v této řadě. Zvolen byl modul s čipem XC7K325T pro svoji velikost. Přesná specifikace modulu i čipu jsou vidět pod následujícími obrázky.



Obr. 23: Architektura modulu Mercury KX1 [27]



Obr. 24: Modul KX1 [27]

Specifikace čipu XC7K325T [28]:

Pracovní teplota:	0 až 85 °C
Typické provozní napětí:	1 V
Logické buňky:	326 080
DSP Slices:	840
Paměť:	16 020 kbit
GTX Transceivery:	16
I/O piny:	500

Specifikace modulu KX1 [27]:

Pracovní teplota	0 až 85 °C
2x Gigabitový Ethernet	
Rozhraní USB 3.0	
4x PCIe 2.0	
2x paměť RAM typu DDR3	(2048 MB a 512 MB)
Flash paměť	64 MB
158 vstupně/výstupních pinů	
Napájecí napětí:	5 až 15 V
4x LED diody	

2.1.3 AD převodník AD9680-500

Pro digitalizaci reálného signálu je použit převodník AD9680 se vzorkovací rychlostí 500 MS/s. Obvod AD9680 obsahuje dva 14bitové převodníky a rozsah měřeného napětí je 2,06 Vpp. Komunikace s FPGA probíhá po vysokorychlostní sériové sběrnici se standardem JESD204B a obsahuje 4 vysokorychlostní linky. Dále obsahuje SPI pro konfiguraci, děličky vzorkovacího kmitočtu nebo programovatelnou zakončovací impedanci. Další parametry:

SNR _{170MHz} :	69,0 dBFS
ENOB _{170MHz} :	11,1 bitů
SFDR _{170MHz} :	88 dBFS
Šířka pásma:	2 GHz

JESD204 je nový vysokorychlostní standard pro sériové linky. Tento standard ve variantě JESD204B dokáže přenášet data s rychlostí až 12,5 GS/s na jednu linku. Rychlost linky je dána vzorcem na následující straně. [30], [31]

$$LLR = \frac{M \cdot N \cdot \left(\frac{10}{8}\right) \cdot f_{OUT}}{L} \quad (17)$$

M ... počet převodníků

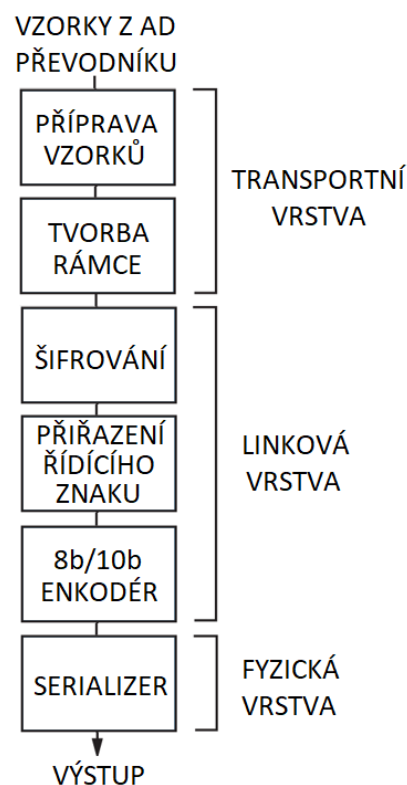
N ... rozlišení převodníku zaokrouhlené na oktet nahoru

f_{OUT} ... vzorkovací kmitočet

L ... počet linek

$$LLR = \frac{2 \cdot 16 \cdot \left(\frac{10}{8}\right) \cdot 500 \cdot 10^6}{4} = 5 \text{ GS/s} \quad (18)$$

Standart JESD204 se skládá z transportní, linkové a fyzické vrstvy. Blokové schéma je vidět na následujícím obrázku. Na straně FPGA se jako fyzické rozhraní využívají vysokorychlostní transceivery GTX, za kterými následují linková a transportní vrstva. [13]

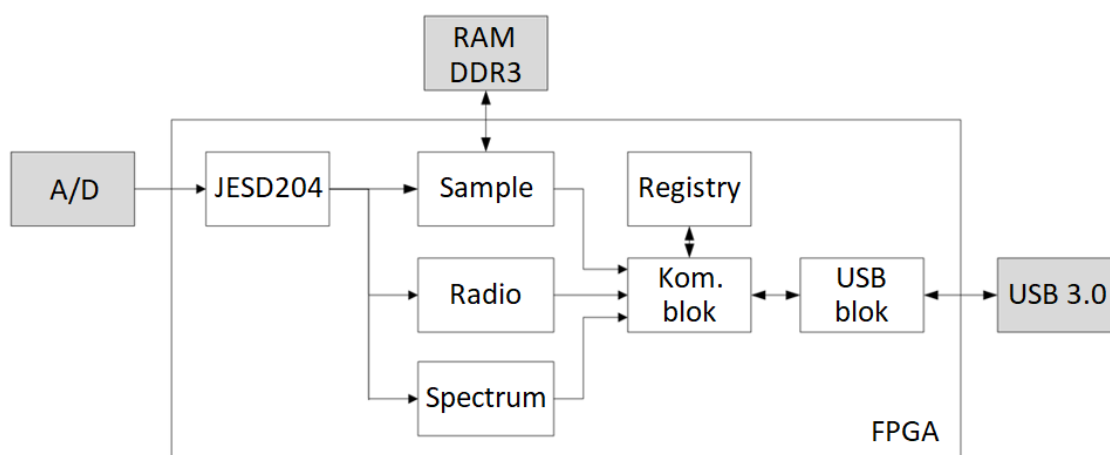


Obr. 25: Vrstvy vysílací části JESD204 v AD převodníku [13]

Jak bylo řečeno, použitý převodník nabízí vzorkovací rychlost 500 MS/s. Ovšem tak vysokou frekvenci již zvolené FPGA neumožňuje využívat. Možností bylo zpracovávat data paralelně na kmitočtu 250 MHz nebo přenastavit převodník a snížit jeho vzorkovací rychlost. Zvolena byla druhá varianta a převodník se využívá na vzorkovacím kmitočtu 250 MS/s.

2.2 Návrh systému

Na obrázku Obr. 26: Blokové schéma navrženého systému je vidět navržený systém, který obsahuje návrh FPGA čipu a několik periférií. Pro komunikaci s počítačem slouží rozhraní USB 3.0, zprostředkujícím obvodem je čip FX3 od společnosti Cypress. Pro získání reálných dat slouží AD převodník a pro uchovávání dat je zde RAM paměť typu DDR3 o velikosti 2 GB. Uvnitř FPGA je navrženo několik základních bloků. Jedná se o 3 bloky, které zajišťují základní režimy (Sample, Spectrum, Radio). Dále je zde blok pro komunikaci s převodníkem k získání vzorkovaných dat. V bloku Registry jsou uložena veškerá nastavení pro jednotlivé režimy a řídicí povely systému. Komunikační blok zajišťuje propojení mezi USB blokem a registry. Dále zajišťuje obsluhu jednotlivých režimů a zpracování jejich výstupů do USB bloku. Ten řeší komunikaci s modulem FX3. Veškeré dílčí části jsou podrobně popsány dále v textu.



Obr. 26: Blokové schéma navrženého systému

Jak již bylo řečeno, pro digitalizaci vstupních dat jsou použity dva extrémně rychlé 14bitové převodníky 500 MS/s se sériovým výstupem (používám 250 MS/s). Možné je takto pomocí každého z nich snímat signály od DC až po frekvenci 125 MHz. Primárně se využívá pouze jeden ze vstupů, ten je využíván všemi třemi režimy. Druhý převodník je napojen pouze na Sample blok a lze takto vzorkovací režim využít jako dvoukanálový osciloskop. Z důvodu 14bitové šířky se vstupní data rozšířila nejvyšším bitem na 16 bitů, a to z důvodu lepší manipulace s daty.

2.2.1 Režimy

Systém nabízí několik základních režimů. Ty jsou zajišťovány jednotlivými bloky, které lze vidět ve schématu výše. Všechny tyto režimy se spouští z počítačové aplikace, možno je jak samostatně, tak i souběžně.

Režim spektra

Prvním režimem je přehled spektra, kdy v aplikaci vidíme frekvenční spektrum. To využívá FFT se šířkou spektrální čáry 3,8 kHz a frekvenční rozsah je od 0 do 125 MHz. Lze zde libovolně nastavit rozmezí kmitočtů a zobrazovat pouze část spektra.

Vzorkovací režim

Dalším režimem je vzorkovací režim. V blokovém schématu je zajištěn Sample blokem. Slouží k navzorkování vstupních dat z převodníku a uložení do paměti. Počítačová aplikace si data poté vyčte a zobrazí. U režimu lze nastavit několik parametrů. Prvním je délka záznamu. Dále Trigger, u kterého lze zvolit podmínky spuštění záznamu (samovolné spuštění, nástupná/sestupná hrana, nebo amplituda). Také lze nastavit předstih Pretrigger, který slouží k nabrání vzorků ještě před spuštěním záznamu.

Úzkopásmový režim

Posledním režimem je Úzkopásmový přijímač zajišťovaný blokem Radio. Cílem tohoto bloku je naladit se na určitý kmitočet s danou šířkou pásma a získaná data poté odesílat kontinuálně do počítače. Z aplikace jdou opět nastavit základní parametry. Prvním je daná frekvence, na kterou se úzkopásmový přijímač naladí, dále šířka pásma nebo druh demodulace. Možností je také zvolit automatické vyhledávání nejsilnějšího signálu. V rozmezí zadaného kmitočtu vyhledá ten s nejsilnější amplitudou a na něj se poté naladí.

2.2.2 Řízení systému

Uvnitř systému je vyhrazen paměťový prostor, který slouží jako přepisovatelné registry. Jejich hodnoty jsou nastavovány z počítače a slouží pro řízení celého systému. Hlavním registrem je registr POVEL, podle zadané hodnoty dochází ke spuštění nebo zastavení některého z režimů. Dále jsou v registrech uloženy různé parametry pro každý z režimů. Blok s registry reaguje na každý zápis a následně nastavuje cílené bloky.

Jak bylo řečeno, nejdůležitějším registrem je POVEL, jedná se o první registr v paměťovém prostoru. Jednotlivé řídicí povely jsou vidět v následující tabulce. Vždy se jedná o spuštění nebo zastavení některého z režimů. Režimy Spektrum nebo Úzkopásmový přijímač odesílají data po spuštění kontinuálně a povel Stop dojde k jejich zastavení. Rozdíl je u Sample režimu, který proběhne pouze jednou. Data se nasbírají a poté se čeká na vyčtení počítačovou aplikací. Nejedná se tedy o kontinuální odesílání a povel Stop by zde nemusel být. Funguje ovšem jako reset, vrací blok do výchozího stavu, také dochází k vymazání fifo paměti. Důvodem je neočekávaný pád aplikace během přenosu.

Tab. 3: Povely pro řízení systému

Hodnota	Význam
0x01	Start sample
0x02	Stop sample
0x03	Start spektrum
0x04	Stop spektrum
0x05	Start úzkopásmového
0x06	Stop úzkopásmového

2.2.3 Řídící registry

Registrů pro řízení je celkem 15 o velikosti 32 bitů. První registr slouží pro řídicí povely a zbylých 14 obsahuje parametry jednotlivých režimů. Jak je vidět v následující tabulce, některé registry jsou nevyužité, to z důvodu rezervy do budoucna nebo pro testování.

Tab. 4: Přehled pregristrů

	Bity							
Adresa	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
0								POVEL
1					T_TYPE	T_SRC	LEN	
2	T_LEVEL				PRET			
3								
4								
5						R_DEM	R_BW	
6	R_F1							
7	R_F2							
8								
9								
10							S_DEC	
11	S_F1							
12	S_F2							
13								
14								
15								

Spektrum

- S_F1 - zobrazovaná počáteční frekvence spektra
- S_F2 - zobrazovaná koncová frekvence spektra
- S_DEC - decimace dat

Úzkopásmový režim

- R_DEM - typ demodulace, 0 = bez, 1 = AM, 2 = FM
- R_BW - šířka pásma
- R_F1 - frekvence pro vyhledávání počáteční
- R_F2 - frekvence pro vyhledávání koncová

Pokud se frekvence u úzkopásmového režimu rovnají, nedojde k vyhledávání, ale ladí se na F1. Trošku zavádějící také je, že píší o kmitočtech, ale vstupní frekvence se zadává podělena 3,8 kHz. Tato hodnota odpovídá šířce spektrální čáry z FFT. Stejně tak se kmitočty v režimu Spektrum zapisují totožným způsobem. Jinak řečeno se zadávají přímo indexy počáteční a koncové spektrální čáry.

Sample režim

LEN	- délka záznamu
T_SRC	- kanál pro trigger
T_TYPE	- typ triggeru, 0 = bez, 1 = level, 2 = nástupná, 3 = sestupná
T_LEVEL	- level triggeru
T_PRET	- délka předstihu

Délka záznamu u Sample režimu je dána následující tabulkou. Do registru se nezapisuje přímo délka, ale pouze kód, který ji interpretuje. Počet 16b vzorků musí být vždy dělitelný 16, důvodem je vstupní port RAM paměti o šířce 256 bitů. Z tohoto důvodu počty vzorků s kódem 0 a 1 nejsou kulaté jako ostatní.

Tab. 5: Přehled možných délek záznamů u vzorkovacího režimu

		2 kanály		
Kód	Délka [ms]	Počet vzorků 16b	Počet slov 32b	Velikost [MB]
0	0,01	4992	2496	0,010
1	0,02	10000	5000	0,020
2	0,05	24992	12496	0,050
3	0,1	50000	25000	0,100
4	0,2	100000	50000	0,200
5	0,5	250000	125000	0,500
6	1	500000	250000	1,000
7	2	1000000	500000	2,000
8	5	2500000	1250000	5,000
9	10	5000000	2500000	10,000
10	20	10000000	5000000	20,000
11	50	25000000	12500000	50,000
12	100	50000000	25000000	100,000
13	200	100000000	50000000	200,000
14	500	250000000	125000000	500,000
15	1000	500000000	250000000	1000,000

2.2.4 Adresní prostor

Adresní prostor je rozčleněn do několika částí a každý režim je rozlišen svou adresou. Registry začínají na adrese 0 a jsou až do adresy 15. U úzkopásmového režimu a spektra je to trochu jinak. K adresám z tabulky jsou ještě přičteny určité hodnoty. U úzkopásmového režimu to je frekvence naladění a u spektra je to index první zobrazované spektrální čáry. U sample režimu začíná adresa na dané hodnotě, ale s každým dalším odchozím balíkem se inkrementuje. Důvodem je velké množství dat, nejdříve dojde k jejich uložení a poté se přenáší po částech.

Tab. 6: Rozčlenění adresného prostoru

Zdroj dat	Počáteční adresa
Registry	0x00000000
Úzkopásmový	0x10000000
Spektrum	0x20000000
Sample	0x30000000

2.2.5 Komunikační blok

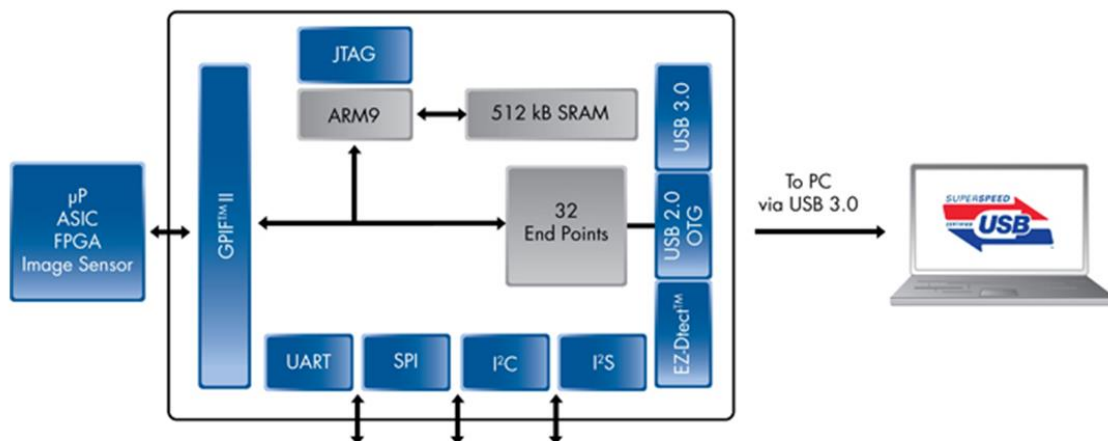
Cílem tohoto bloku je propojit bloky zajišťující jednotlivé režimy, USB a registry. Při příchodu jakýchkoliv dat z USB rozezná daný blok data podle hlavičky a pomocí multiplexorů je správně přepojí. Stejně tak hlídá bloky zajišťující režimy a v případě připravených dat informuje USB blok. Protože USB blok má pouze jeden vstup dat a zdrojů je zde více, komunikační blok opět řídí vyčítání ze správného zdroje multiplexováním.

Pro informaci, že jsou již připravena data má každý blok výstupní port IRQ. Ten značí přerušení, že došlo k naplnění RAM nebo bylo například načteno celé spektrum. Komunikační blok tyto signály sjednotí. V případě příchodu více přerušení najednou, nebo pro zaneprázdnění USB bloku se přerušení ukládá. USB blok si je vyžádá, až nebude zpracovávat žádnou komunikaci.

2.3 Komunikace po USB 3.0

2.3.1 USB 3.0 Cypress FX3

Na následujícím obrázku je vidět blokový diagram čipu FX3. Obsahuje programovatelný procesor ARM9 a pro komunikaci s FPGA GPIF II rozhraní, taktéž programovatelné. Možností tohoto modulu je přizpůsobit si obvod dle svých potřeb pro danou komunikaci. Programování a návrh komunikace probíhá v softwaru EZ-USB FX3 SDK, který je ke stažení na stránkách výrobce.

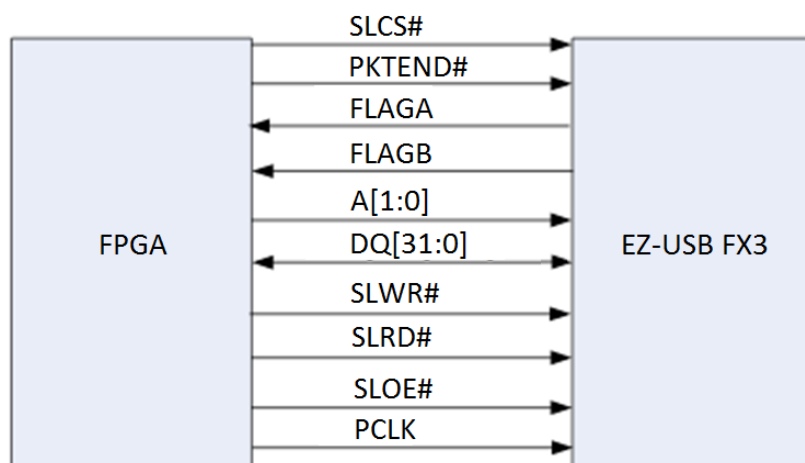


Obr. 27: Architektura čipu Cypress FX3 [33]

Ke stažení jsou také různé příklady použití a připravené soubory pro konfiguraci. V základu lze volit ze 4 druhů komunikace, ale možný je i vlastní návrh. Jednou z možností byl například stream, já ovšem zvolil Slave FIFO Interface. Jedná se o obousměrnou komunikaci mezi FPGA a počítačem využívající buffery. Při příchozích datech dojde k uložení do bufferu a protistrana si následně data vyčte.

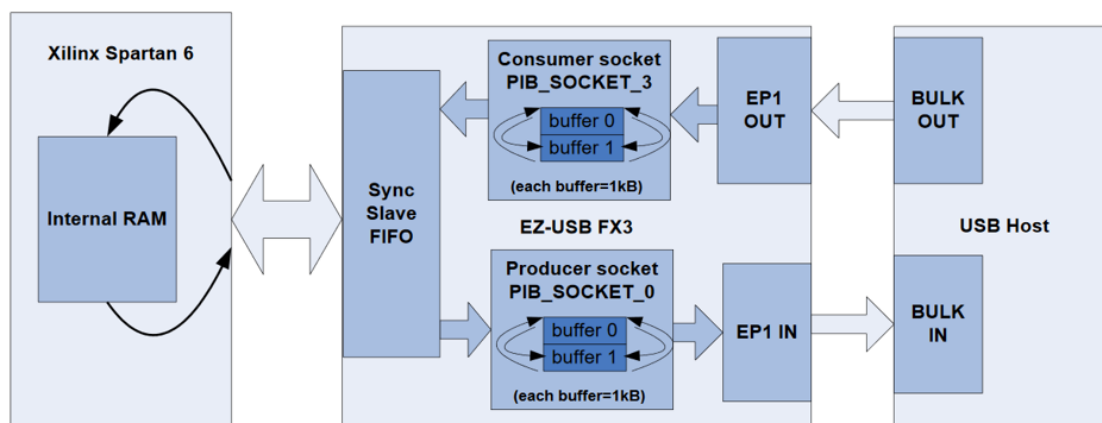
Jak již bylo řečeno, programovatelné je rozhraní GPIF II i ARM procesor. Pro rozhraní GPIF II se navrhuje stavový automat. Ten určuje vstupní a výstupní porty čipu, zajišťuje komunikaci s FPGA a stanovuje časové parametry a průběhy komunikace. V ARM procesoru se nastavují různé vstupní a výstupní buffery, jejich hloubka, velikost nebo počet vláken pro přenos.

Na následujícím obrázku lze vidět vstupy a výstupy modulu FX3 při použití Slave FIFO Interface. Jedná se o hodinový signál, přenášená data, různé povolující signály, příznaky FLAG, zda jsou v bufferu data nebo je připravený pro zápis. Časové diagramy i přesný popis jednotlivých pinů pro zvolený případ jsou k nahlédnutí v katalogovém listu.



Obr. 28: Propojení FPGA a čipu FX3 při režimu Slave Fifo Interface [32]

Níže na obrázku je zobrazena struktura pro komunikaci vytvořená v procesoru ARM. Opět je použit vzor od výrobce. Využívá dvě vlákna, každé pro komunikaci jedním směrem. Vytvořené cesty obsahují buffery, ke kterým náleží příznaky FLAGA a FLAGB. Počet bufferů i jejich velikost lze snadno rozšířit změnou ve vzorovém kódu, který je psán v jazyce C#. Počet bufferů byl zvýšen na 8 a velikost na 10 kB. I když má obvod primárně sloužit pro komunikaci s USB 3.0, lze komunikovat i s portem USB 2.0. V tomto případě je ovšem použitelná velikost bufferů poloviční.



Obr. 29: Vnitřní struktura čipu FX3 při režimu Slave Fifo Interface [32]

Výhodou modulu také je přiložená počítačová aplikace. Dodány jsou i zdrojové kódy psané v jazyce C# a taktéž knihovny. Z počátku se aplikace hodí pro testování komunikace s FPGA a lze ji následně upravit pro vlastní potřebu.

2.3.2 Komunikační protokol

Navržený komunikační protokol je zobrazen v následující tabulce. Odesílají se slova o délce 32 bitů, z nichž první dvě tvoří hlavičku a za nimi mohou následovat data. První slovo je tzv. Řídící. Obsahuje kontrolní část, která určuje, o jaká data nebo činnost se jedná. Dále obsahuje počet odesílaných dat. Druhé slovo obsahuje adresu, ze které data pochází, nebo na kterou se má zapisovat. Další slova už náleží datům.

Tab. 7: Protokol komunikace po USB

		Bity			
Slovo		31:24	23:16	15:8	7:0
1.	Řídící	nevyužito	délka		kontrolní
2.	Adresa	adresa			
3. - N.	Data	data			

Tab. 8: Možné hodnoty pro kontrolní Byte

Hodnota	Význam
0x10	Čtení z paměti (Registry, RAM)
0x20	Zápis do paměti (Registry)
0x30	Data – úzkopásmová
0x40	Data – spektrum
0x50	RAM byla naplněna (Příznak bez dat)

2.3.3 Formáty odesílaných dat

Každý režim má formát odesílaných dat mírně odlišný. V následujícím textu jsou formáty dat pro jednotlivé režimy přesně popsány.

Úzkopásmový režim

U úzkopásmového režimu se odesílají data ve dvou formátech. Pokud jsou data bez demodulace, odesílají se vzorky I a Q. Při použití demodulace je pak počet vzorků poloviční a odesílají se přímo za sebou. Při náhledu do tabulky si lze všimnout adresy x"1XXXXXXX", za písmena X je doplněna frekvence, kde je úzkopásmový konvertor naladěn.

Tab. 9: Formát dat pro úzkopásmový režim bez demodulace

	Bity			
Slovo	31:24	23:16	15:8	7:0
1.	x“00“	x“YYYY“		x“30“
2.	x“1XXXXXXXX“			
3.	q0		i0	
4.	q1		i1	
5.	q2		i2	

Tab. 10: Formát dat pro úzkopásmový režim s demodulací

	Bity			
Slovo	31:24	23:16	15:8	7:0
1.	x“00“	N		x“30“
2.	x“1XXXXXXXX“			
3.	am/fm1		am/fm0	
4.	am/fm3		am/fm2	
5.	am/fm5		am/fm4	

Režim spektra

V následující tabulce lze vidět formát odesílaných dat pro režim spektra. Čáry se odesílají přímo za sebou. Při náhledu do tabulky si lze opět všimnout adresy x"2XXXXXXX", písmena X jsou nahrazena indexem první zobrazované spektrální čáry.

Tab. 11: Odesílání dat spektra

	Bity			
Slovo	31:24	23:16	15:8	7:0
1.	x"00"	x"YYYY"		x"40"
2.	x"2XXXXXXX"			
3.	cara1		cara0	
4.	cara3		cara2	
5.	cara5		cara4	

Sample režim

U vzorkovacího režimu se odesílá pouze příznak, že byla naplněna RAM, ale délka je nulová. Aplikace si poté vyčítá data sama, ze zvolené délky ví, kolik dat má vyčíst. Vzorky A odpovídají jednomu převodníku a vzorky B druhému.

Tab. 12: Odeslání příznaku u vzorkovacího režimu

	Bity			
Slovo	31:24	23:16	15:8	7:0
1.	x"00"	x"0000"		x"50"
2.	x"30000000"			

Tab. 13: Vyčítání dat z paměti RAM

	Bity			
Slovo	31:24	23:16	15:8	7:0
1.	x"00"	x"YYYY"		x"10"
2.	x"30000000"			
3.	vz_B_0		vz_A_0	
4.	vz_B_1		vz_A_1	
5.	vz_B_2		vz_A_2	

2.3.4 USB blok

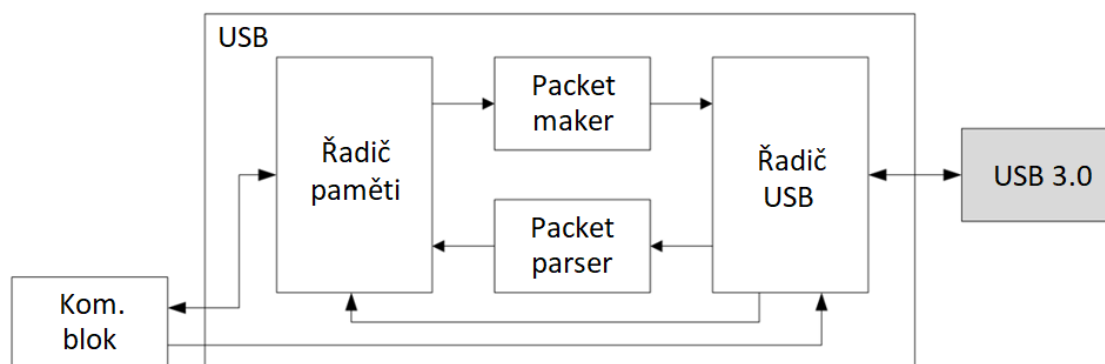
Jak bylo napsáno výše v kapitole Návrh systému, USB blok řeší komunikaci mezi blokem komunikačním a čipem FX3. Skládá se z několika částí a schéma je vidět na obrázku Obr. 30.

Nejdůležitějším je Řadič USB, který komunikuje přímo s čipem dle daných specifikací v katalogovém listu. Řízen je stavovým automatem a výstupem z něj jsou přijatá jednotlivá slova o délce 32 bitů. Dále také zpracovává příznak IRQ z komunikačního bloku a dává příkaz řadiči paměti pro přípravu dat k přenosu do počítače.

Packet parser má za úkol zpracovat přijatou hlavičku, tzn. oddělit délku dat, kontrolní část a adresu. Tyto informace předává již paralelně do řadiče paměti a za nimi následují data.

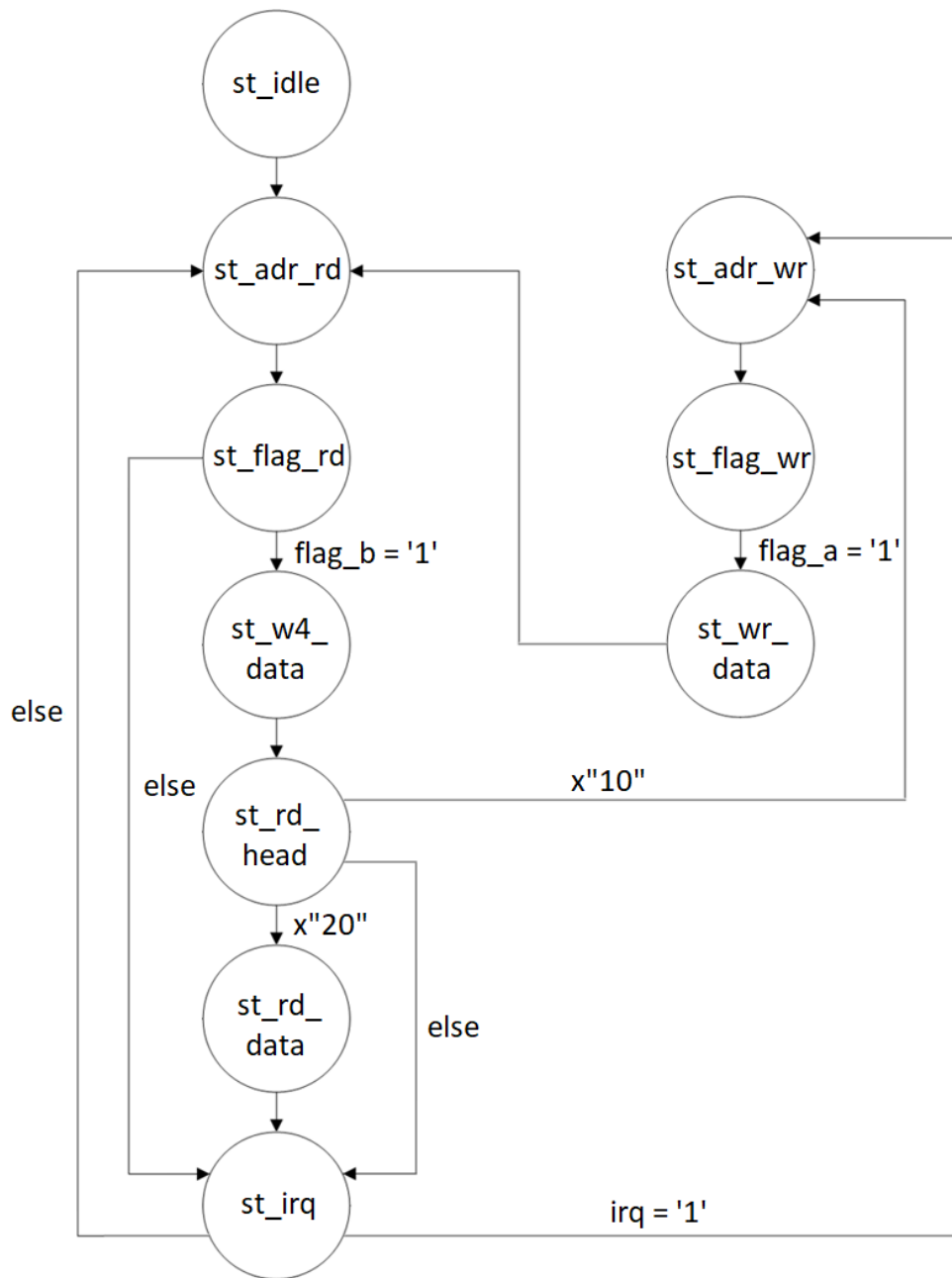
Řadič paměti slouží pro propojení s komunikačním blokem. Obsahuje porty pro komunikaci s pamětí (Write Enable, Read Enable, Adresa, Data). Přijatou hlavičku dekoduje a nastavuje podle ní příslušné signály. Druhou možností je reakce na IRQ z řadiče USB, kdy následuje vyčítání dat z paměti.

Packet maker je opakem parseru, z adresy, délky nebo kontrolního Bytu vytváří opět 32 bitová slova a odesílá je USB Řadiči. Dá se o něm mluvit jako o serializeru.



Obr. 30: Blokové schéma USB bloku

Jak již bylo řečeno, USB obsahuje dvě vlákna, každé pro komunikaci jedním směrem. Tyto vlákna jsou odlišena adresou a komunikace současně není možná. Na obrázku Obr. 28 (propojení USB čipu s FPGA) jsou zobrazeny vodiče FLAGA a FLAGB. Každý z těchto příznaků náleží jednomu vláknu a určuje, zda jsou buffery v cestě plné nebo prázdné. Komunikace probíhá tak, že se zvolí adresa vlákna, zkontroluje se příznak, zda je možné zapisovat nebo vyčítat a provede se přenos dat.



Obr. 31: Stavový automat řadiče USB

st_adr_rd

- je prvním stavem a nastaví adresu vlákna směřovaného z PC do FPGA

st_flag_rd

- v tomto stavu dochází k vyčtení příznaku, zda jsou v bufferu připravená data, pokud ano, nastaví vyčítání dat ve stavu st_w4_data, jinak jde na přerušení IRQ

st_rd_head

- zde dochází k vyčtení a dekódování hlavičky, následují dvě možnosti, buď chce aplikace zapisovat data do FPGA, čemuž odpovídá hodnota kontrolní části x"20", druhou

možností je čtení dat z čipu FX3, čemuž odpovídá hodnota x“10“

st_rd_data

- zde dojde k vyčtení dat a odeslání do packet parseru

st_adr_wr

- do tohoto stavu se dá dostat, pokud přijde kontrolní část s hodnotou x“10“, nastavuje se vlákno pro zápis do počítače

st_flag_wr

- vyčte příznak, zda je buffer prázdný a připravený pro zápis

st_wr_data

- v tomto stavu dojde k vyčítání hodnot z paměti a zápis do USB

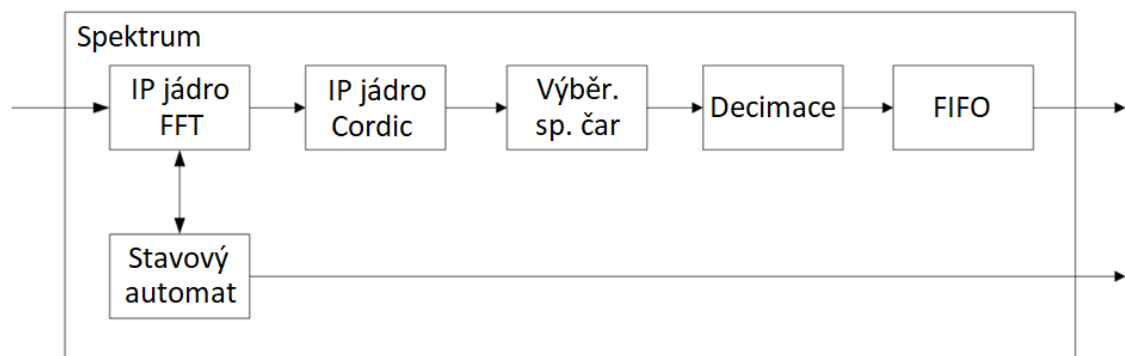
st_irq

- do stavu st_irq se dá dostat, pokud neprobíhá žádná komunikace, kontroluje, zda přišlo přerušení IRQ, případně nastává vyčítání dat z paměti a odesílací sekvence

2.4 Bloky jednotlivých režimů

2.4.1 Blok pro režim Spectrum

Režim spektra je zajišťován blokem Spectrum, ten obsahuje opět několik dílčích částí. Nejdůležitější částí je IP jádro FFT, které má vstupní data z AD převodníku a výstupem jsou poté jednotlivé spektrální čáry. Protože výstupem z FFT jádra jsou reálná a imaginární složka signálu, je třeba z nich vypočítat absolutní hodnotu. K tomu je využito jádro IP Cordic, z něj již vystupuje amplituda spektrální čáry. Následuje výběr spektrálních čar, pokud index čáry spadá do zadaného intervalu, data pokračují dále na decimaci. Decimace z daného počtu vzorků vybírá vždy ten s nejvyšší amplitudou a data zapisuje do výstupního fifa. IRQ přerušení je odesláno stavovým automatem v okamžiku, kdy bylo prozkoumáno celé spektrum, dále čeká na vyprázdnění fifa a opět spouští FFT jádro.



Obr. 32: Blokové schéma pro režim Spectrum

FFT jádro se zvolilo největší, co bylo k dispozici, a to o velikosti 65536. Kvůli vzorkovacímu teorému je využitelných pouze 32768 spektrálních čar. Při vzorkovacím kmitočtu 250 MHz vychází rozlišení spektrální čáry 3,8 kHz.

Decimace

Důvodem decimace bylo zrychlení přenosu spektra do počítače. Při úvaze, že celé spektrum obsahuje 32768 čar a monitor s rozlišením 4K nabízí maximální zobrazení 3840 pixelů, není potřeba přenášet celé spektrum, ale stačí pouze jeho desetina. Tím se ušetří množství přenesených dat.

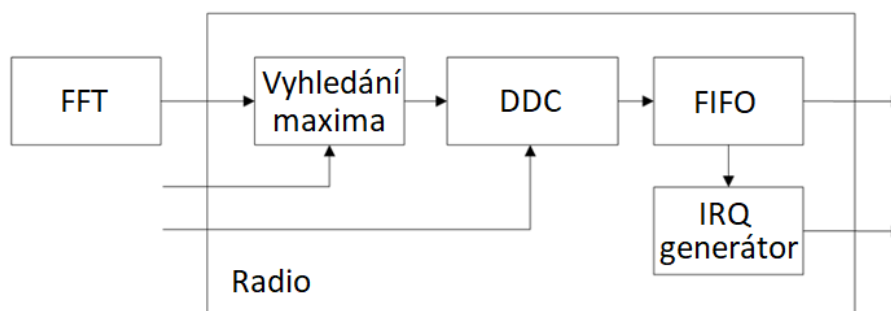
Možností decimace bylo více. Jednou z variant bylo použití nějakého decimačního filtru, například CIC. Druhou možností se nabízelo data průměrovat, což ovšem data značně zkreslovalo. Zvolena byla varianta nejvyšší amplitudy, kdy se z N čar vybere ta s nejvyšší hodnotou. Při sledování spektra беру jako ty nejzajímavější signály ty nejsilnější, poté je možno si spektrum přiblížit.

Spuštění režimu

Daný blok se nespouští žádným start signálem, stavový automat skenuje spektrum neustále. To z důvodu, že jeho výstup může být využíván i úzkopásmovým režimem. Na základě zapnutí režimu spektra v registrech se pouze blokuje nebo povoluje zápis do výstupního fifa a přerušení IRQ.

2.4.2 Blok pro Úzkopásmový režim

Cílem úzkopásmového režimu je naladit se na daný kmitočet se zadanou šířkou pásma. Druhou možností je automatické vyhledání nejsilnějšího signálu v kmitočtové vymezeném intervalu a následně se na něj naladit. Vzorkovaná data jsou poté odesílána kontinuálně do počítače. Bylo zvoleno celkem šest šířek pásma a přesně jsou vypsány dále v textu. Aby bylo možné dosáhnout požadované šířky, je využito jak filtrů FIR, tak i decimačních filtrů, čímž se podstatně snižuje vzorkovací frekvence a tím i datový tok do počítače. Lze provádět kontinuální přenos. Možností je využití demodulace AM nebo FM a lze tento režim použít ve funkci rádia.

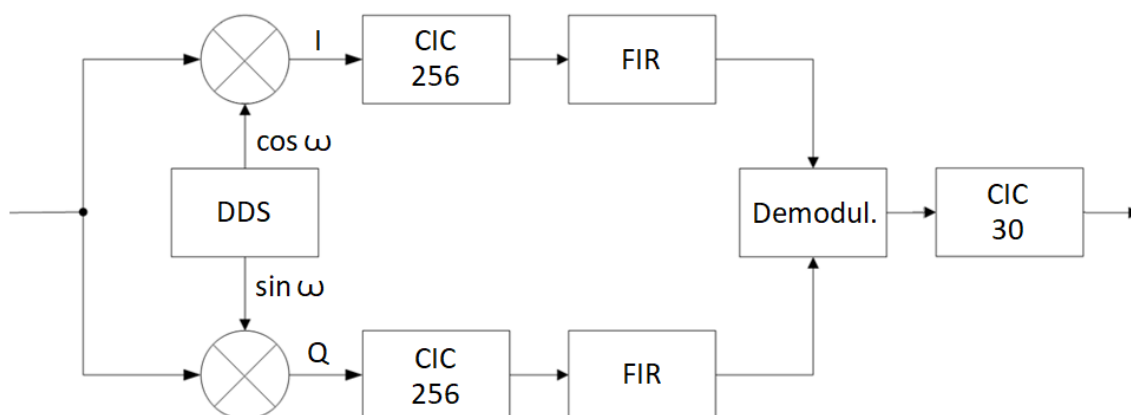


Obr. 33: Blokové schéma pro Úzkopásmový režim

Na předchozím obrázku je vidět blokové schéma pro zmiňovaný režim. Prvním blokem je vyhledání maxima. To využívá spektrální čáry z FFT, porovnává jejich indexy

se zadaným intervalem a souběžně porovnává amplitudy. Výstupem bloku je kmitočet odpovídající indexu spektrální čáry s nejvyšší amplitudou. Druhou variantou je ladění na přesně zadaný kmitočet. V tom případě se ignoruje vyhledávání a do následujícího bloku jde přímo zadaný kmitočet.

Následujícím blokem je DDC neboli digitální down konvertor rozšířený o demodulátory AM a FM. Víceméně celé DDC je tvořeno IP jádrem. Blokové schéma lze vidět na následujícím obrázku.

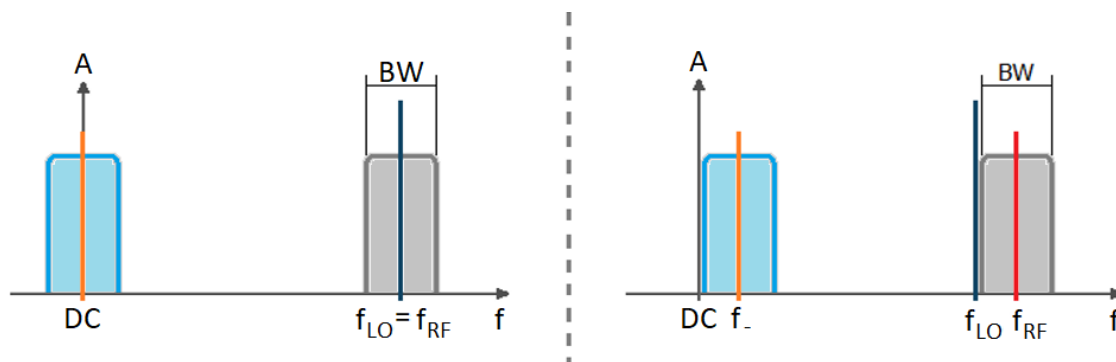


Obr. 34: Blokové schéma DDC

Při nahlédnutí na obrázek si lze všimnout dvou stupňů decimace. První filtr CIC má nastavenou decimaci na hodnotu 256 a snižuje vzorkovací frekvenci na kmitočet 976 kHz. Díky nižšímu kmitočtu lze filtry FIR navrhnout se strmější charakteristikou. Filtr FIR je navržen jako dolní propust a je přepínatelný mezi několika charakteristikami. Ty odpovídají vybraným šířkám pásem a k nahlédnutí jsou dále v textu. Za první filtrací probíhá demodulace, která může být AM nebo FM. Dále následuje znovu snížení vzorkovacího kmitočtu, a to na $f = 32,5$ kHz. Cílem bylo přiblížit se kmitočtu 32 kHz, který odpovídá zvukovému záznamu.

Ladění DDS

Na následujícím obrázku je zobrazeno směšování pomocí DDS. Podle zvolené demodulace se k ladění přistupuje odlišně. Levý obrázek odpovídá amplitudové modulaci a pravý kmitočtové.



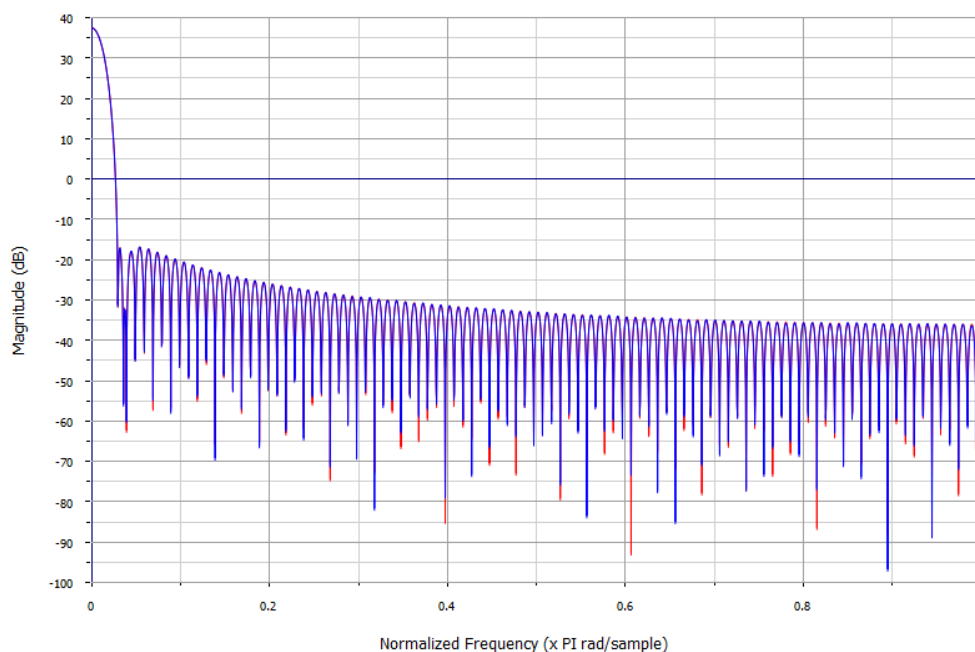
Obr. 35: Princip ladění podle zvolené demodulace, vlevo AM, vpravo FM

U amplitudové demodulace mohou směřovat přímo do základního pásma, kdy demodulace není ovlivněna. Rozdíl je ovšem u FM, kde je potřeba se odladit tak, aby pásmo nezasahovalo přes základní pásmo. Důsledkem by byla deformace směřovaného signálu, která by znemožnila výpočet fáze a tím i demodulaci. Cílem je se vždy odladit o polovinu zvolené šířky pásma. Přesné hodnoty i se zvolenými šířkami jsou vidět v následující tabulce.

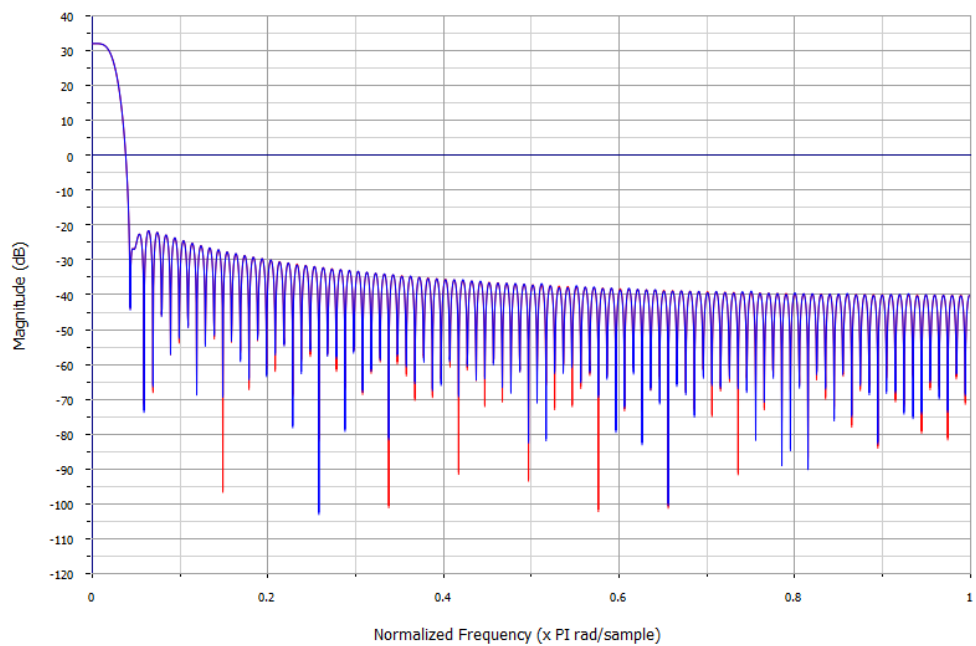
Tab. 14: Nastavení jednotlivých filtrů pro dané šířky pásma a odladění pro FM

index [-]	f0 filtru [kHz]	Šířka pásma AM [kHz]	Šířka pásma FM [kHz]	Odladění FM vypoč. [kHz]	Odladění FM real. [kHz]
0	6,25	12,5	6,25	3,125	3,81
1	12,5	25	12,5	6,25	7,63
2	25	50	25	12,5	11,44
3	50	100	50	25	26,70
4	100	200	100	50	49,59
5	200	400	200	100	99,18

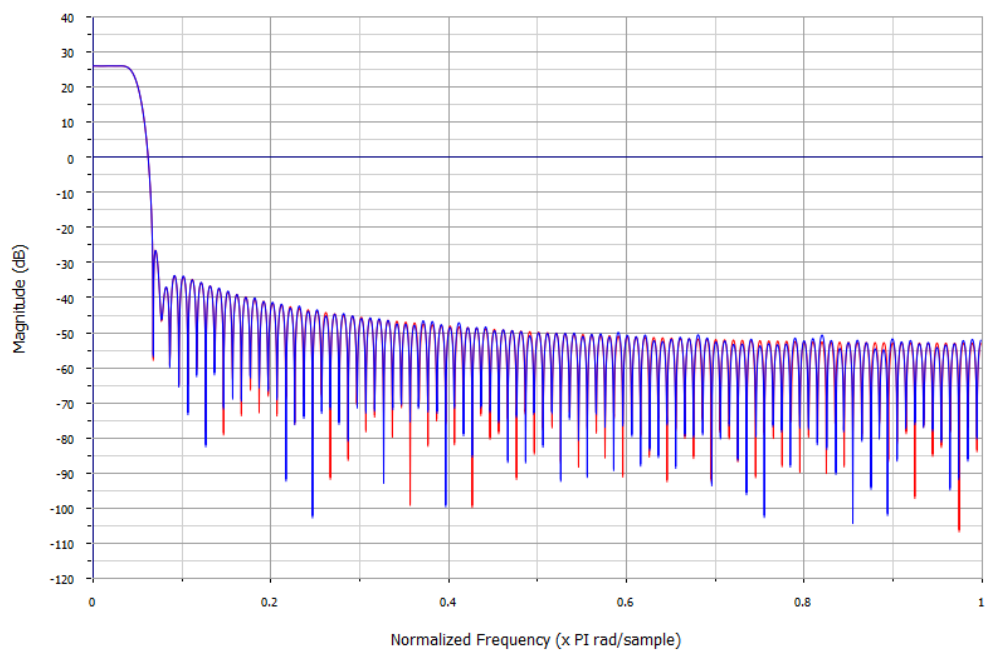
Při náhledu do tabulky si lze všimnout, že reálné odladění neodpovídá přesně teoretickému. Tato hodnota totiž závisí na zvoleném kroku syntezátoru DDS. Pro snazší řízení byl krok DDS zvolen na 3,8 kHz, což odpovídá šířce spektrální čáry z bloku FFT. Při automatickém vyhledávání lze totiž využít indexu čáry z FFT a přímo ho napojit do syntezátoru. Reálná odladění jsou tedy nejbližší teoretickým při zvolené přesnosti. Dále si lze v tabulce všimnout indexu. Místo šířky pásma se totiž odesílá kód, který ji interpretuje. Každému indexu neboli šířce pásma také odpovídá jiná charakteristika FIR filtru. Navržené filtry jsou zobrazeny na následujících obrázcích.



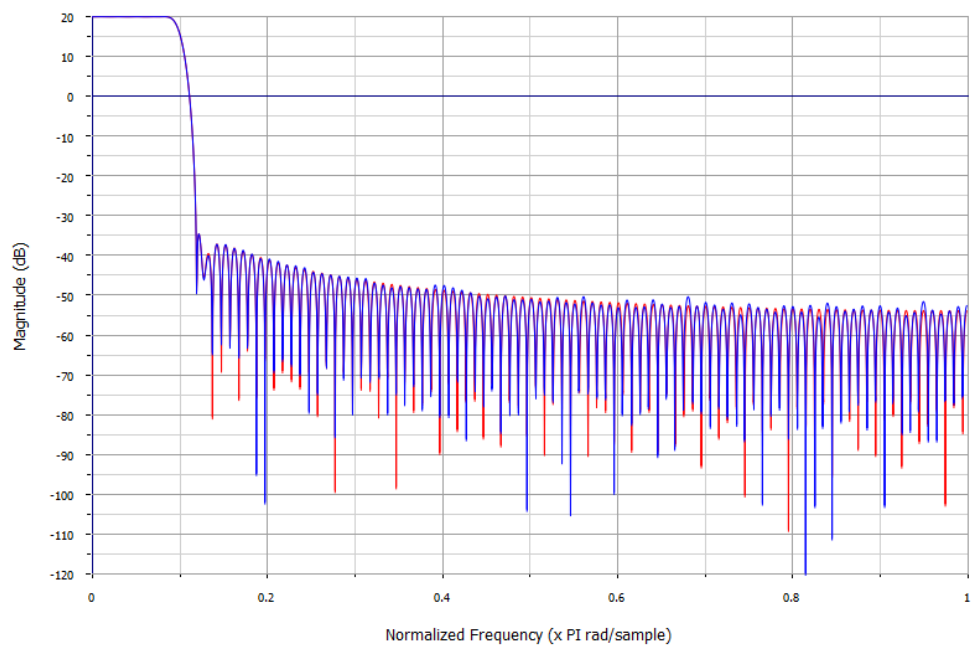
Obr. 36: Navržený filtr typu DP 6,25 kHz



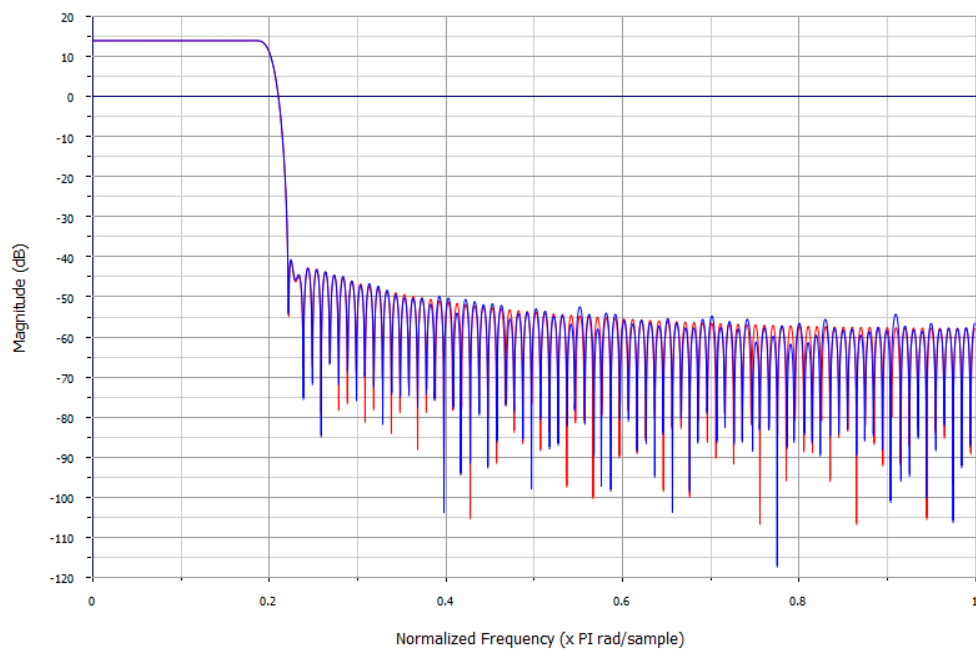
Obr. 37: Navržený filtr typu DP 12,5 kHz



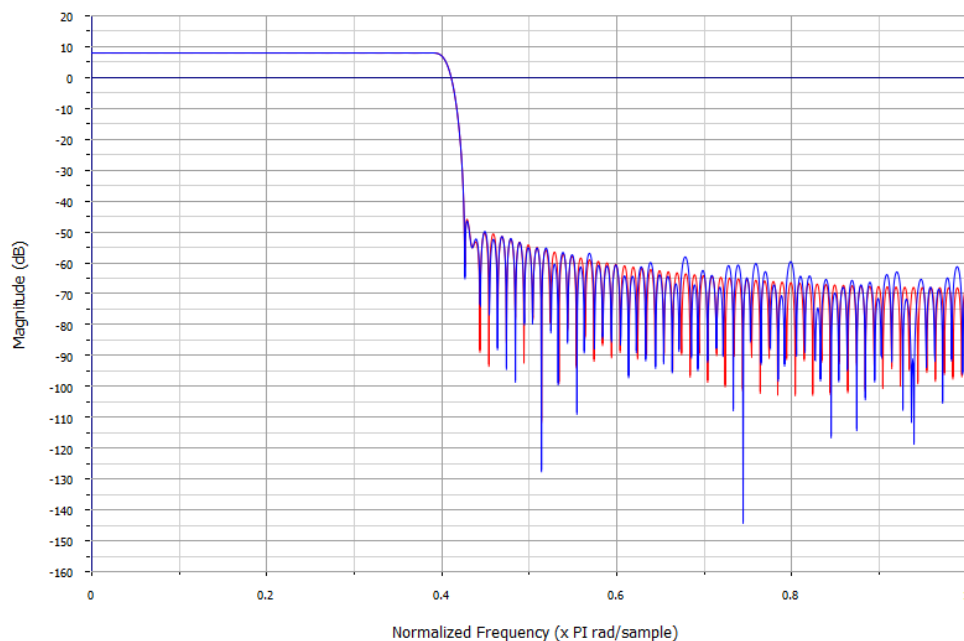
Obr. 38: Navržený filtr typu DP 25 kHz



Obr. 39: Navržený filtr typu DP 50 kHz



Obr. 40: Navržený filtr typu DP 100 kHz



Obr. 41: Navržený filtr typu DP 200 kHz

Demodulace AM

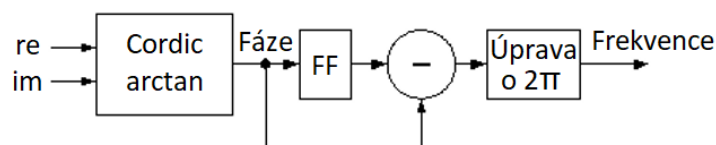
Pro demodulaci AM je využito IP jádro Cordic. Jeho vstupy jsou reálná a imaginární složka z DDC bloku. Výstupem je pak absolutní hodnota těchto kartézských souřadnic. Ta odpovídá amplitudě modulačního signálu.

Demodulace FM

Pro kmitočtovou demodulaci je využito fázového diskriminátoru. Opět je použito jádro Cordic, které tentokrát počítá z komplexních hodnot fázi pomocí funkce arcTan. Následující vztah popisuje závislost mezi fází a komplexními hodnotami.

$$\theta = \arctan\left(\frac{im}{re}\right). \quad (19)$$

Fáze je dále zpožděna přes klopný obvod a je vypočítán rozdíl dvou za sebou jdoucích vzorků. Protože rozsah výstupní fáze je $\pm\pi$, je nutné podle znaménka a velikosti fáze připočítat ještě hodnotu 2π . Vypočtený rozdíl fází je již úměrný frekvenci.



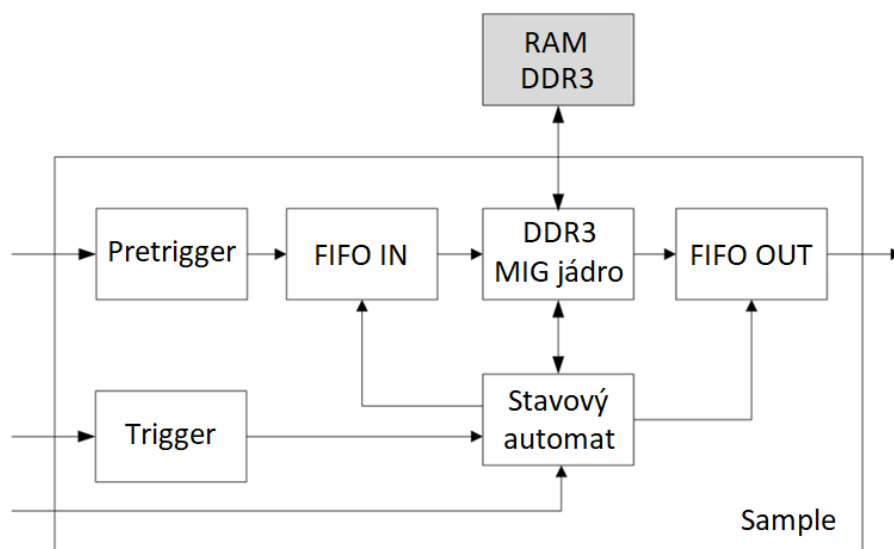
Obr. 42: Fázový diskriminátor

Výstupem z DDC bloku jsou již filtrovaná data s danou šířkou pásma a přímo se zapisují do výstupního fifu.

Posledním blokem je IRQ generátor. Jedná se o stavový automat, který slouží ke generování příznaku IRQ na základě počtu zapsaných dat ve výstupním fifu. Počet je nastaven na 512 16bitových dat, což je 256 slov. Po vystavení přerušení následně počítá vyčítaná data. V okamžiku, až se výčet rovná hodnotě 256, dojde k navrácení do prvního stavu, kde opět hlídá počet dat ve fifu.

2.4.3 Blok pro Sample režim

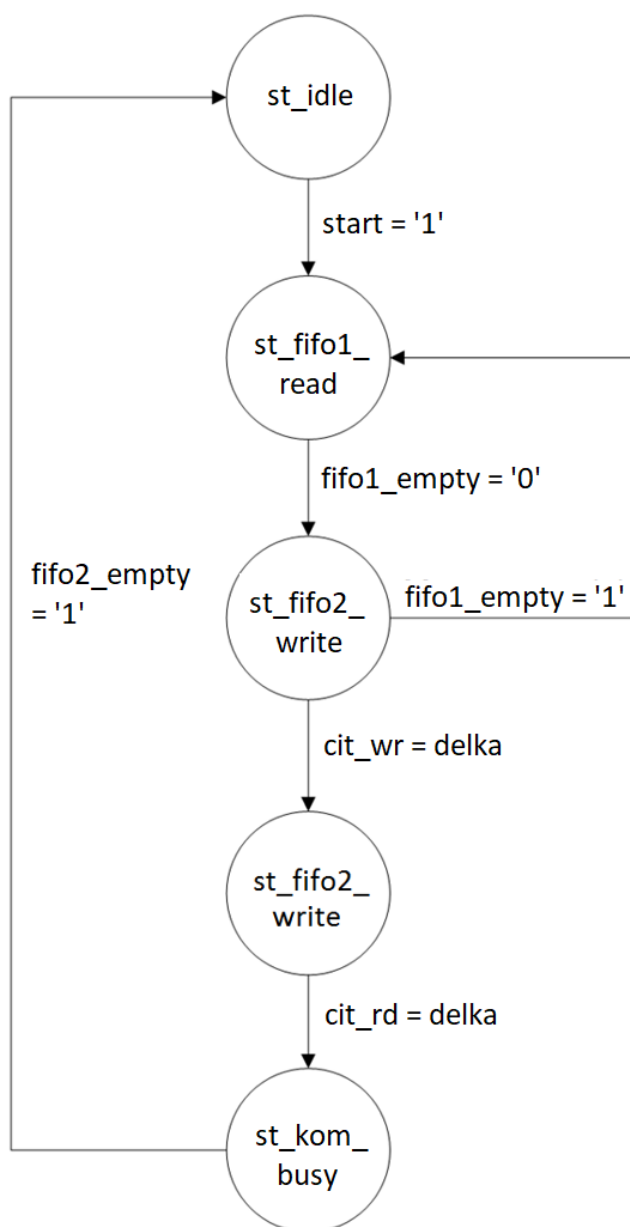
Tento blok slouží pro obsluhu Sample režimu, jinak řečeno vzorkovacího režimu. Na obrázku Obr. 43 je vidět blokové schéma. Pro komunikaci s fyzickou DDR3 pamětí bylo vygenerováno IP jádro MIG. Veškerou obsluhu zde zajišťuje stavový automat. Jako pretrigger je použit posuvný registr a velikost je volena signálem PRET z řídicích registrů. Vstupní a výstupní fifo paměti jsou zde ze dvou důvodů. Prvním důvodem je bitová šířka dat, která je 16 bitů, ale MIG jádro má vstup 256 bitů. Data se nabírají ve fifu a až poté dochází k zápisu do RAM paměti. Druhý důvod je oddělení hodinových signálů. Sample blok pracuje s hodinovým signálem z MIG jádra, vstupní data jsou ovšem na hodinách převodníků a výstupní na hodinách USB. Blok Trigger je spuštěn řídicím signálem z registrů a dále spouští stavový automat.



Obr. 43: Blokové schéma Sample bloku

Stavový automat RAM

Tento stavový automat zajišťuje zpracování dat v Sample režimu. Vstupem mu je počet dat, která má uložit a řídicí signál start. Po spuštění start signálem kontroluje vstupní fifo. Jakmile jsou v něm data připravena, provádí vyčtení, následné zapsání do RAM paměti a veškerá data počítá. Toto opakuje, dokud nedojde ke shodě s požadovaným počtem. Poté dochází k vyčítání z RAM a dojde k zápisu do výstupní fifo paměti. V tomto okamžiku se odešle příznak IRQ do komunikačního bloku a kontroluje se počet dat ve výstupním fifo. Postupně se fifo doplňuje, dokud není RAM paměť prázdná (počet vyčtených dat má shodu). Proces Sample režimu končí v okamžiku, kdy je i výstupní fifo prázdné, což značí vyčtení veškerých dat. Následně se automat vrací do výchozího stavu.



st_idle

- kontroluje signál start

st_fifo1_read

- kontroluje, zda fifo není prázdné, vyčítá z něj data a poté pokračuje na zápis do paměti RAM

st_ram_write

- provádí zápis do RAM a porovnává počet zapsaných dat s danou délkou, poté jde buď na zápis do výstupního fifa, nebo čekat do předchozího stavu

st_fifo2_write

- provádí vyčtení z RAM a zápis do výstupního fifa, odesílá IRQ příznak
- porovnává počet vyčtených dat z RAM s požadovanou délkou, v případě shody pokračuje do dalšího stavu

st_kom_busy

- čeká, dokud není výstupní fifo vyčtené

Obr. 44: Stavový automat pro řízení RAM

Funkce Trigger

Funkce Trigger umožňuje 4 typy spouštění záznamu:

- Samovolné spuštění
- Level
- Nástupná hrana
- Sestupná hrana.

Tato komponenta obsahuje dva důležité vstupy, jedním jsou data pro záznam a druhým je podmínka Level. Nejdříve dochází k porovnání těchto hodnot, a pokud jsou data vyšší než Level, dochází k nastavení příznaku na hodnotu '1'. Dále se kontroluje tento příznak. Pokud došlo ke změně z '0' na '1' nebo naopak, dochází k nastavení příznaků pro náběžnou nebo sestupnou hranu. Samotný trigger je poté řízen stavovým automatem, který obsahuje pouze tři stavy. Spuštění nastává hodnotou z registrů. Ve druhém stavu čeká na zmíněné příznaky podle zvoleného typu spuštění. A ve třetím stavu čeká na konec záznamu.

Pretrigger

Pretrigger slouží pro nabrání určitého počtu vzorků ještě před samotným spuštěním záznamu. Funkci pretriggeru zajišťuje posuvný registr o velikosti 4096 buněk, což odpovídá 16 μ s záznamu. Podle zvolené délky předstihu se zvolí buňka na výstup, a její obsah je připojen následně do RAM.

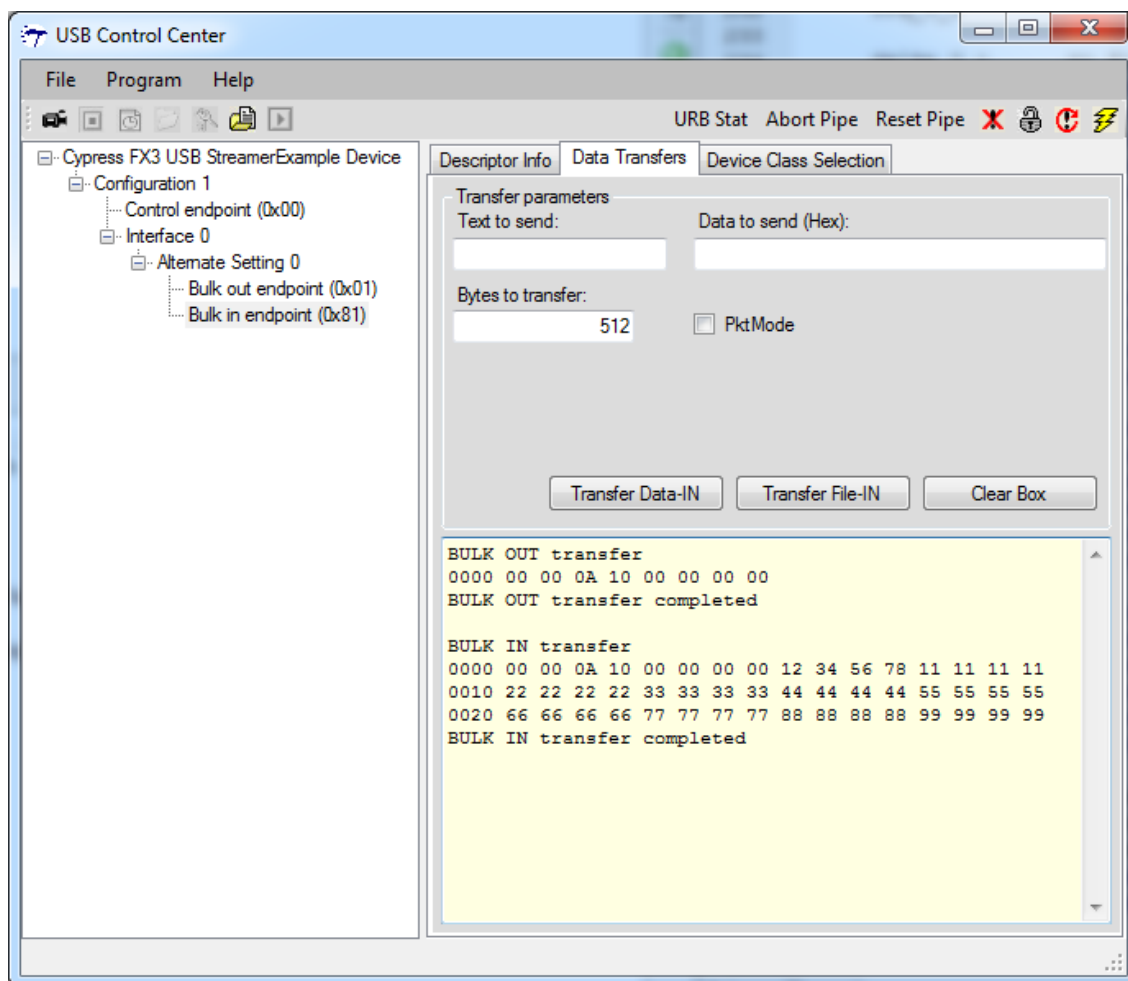
2.5 Oživení zařízení

V této části práce budou zobrazeny pořízené průběhy případně obrázky při testování a ožívování zařízení. Konkrétně se jedná o demodulaci signálu u úzkopásmového režimu, testování komunikace po USB, měření charakteristiky FFT nebo obsluha DDR3 paměti.

2.5.1 USB komunikace

Před samotným ožívováním USB čipu probíhalo testování pomocí simulací a porovnávání s průběhy z technického listu. Po úspěšných simulacích se přešlo k samotnému oživení čipu FX3, ten obsahoval naflashovaný bootloader. Pro testování a přehrání čipu byl dodán software Control Center, pomocí kterého bylo možné nahrát nový firmware. Ovšem po prvním přehrání dojde ke ztrátě bootloaderu a není možný další přepis. Výrobci tento problém ošetřili odporem R1600, který je potřeba při zapnutí zařízení vyzkratovat a v Control Center se objeví opět bootloader.

Obrázek Obr. 45 zobrazuje aplikaci Control Center použitou pro test komunikace s FPGA. V hradlovém poli bylo vytvořeno deset registrů naplněných různými hodnotami od adresy 0 až po adresu 9. Test proběhl následovně. Nejdříve došlo k odeslání příkazu pro vyčtení deseti hodnot od adresy X"00000000" pomocí "Bulk out endpoint". Poté došlo k vyčtení bufferu USB přes "Bulk in Endpoint", kde již data byla připravena.



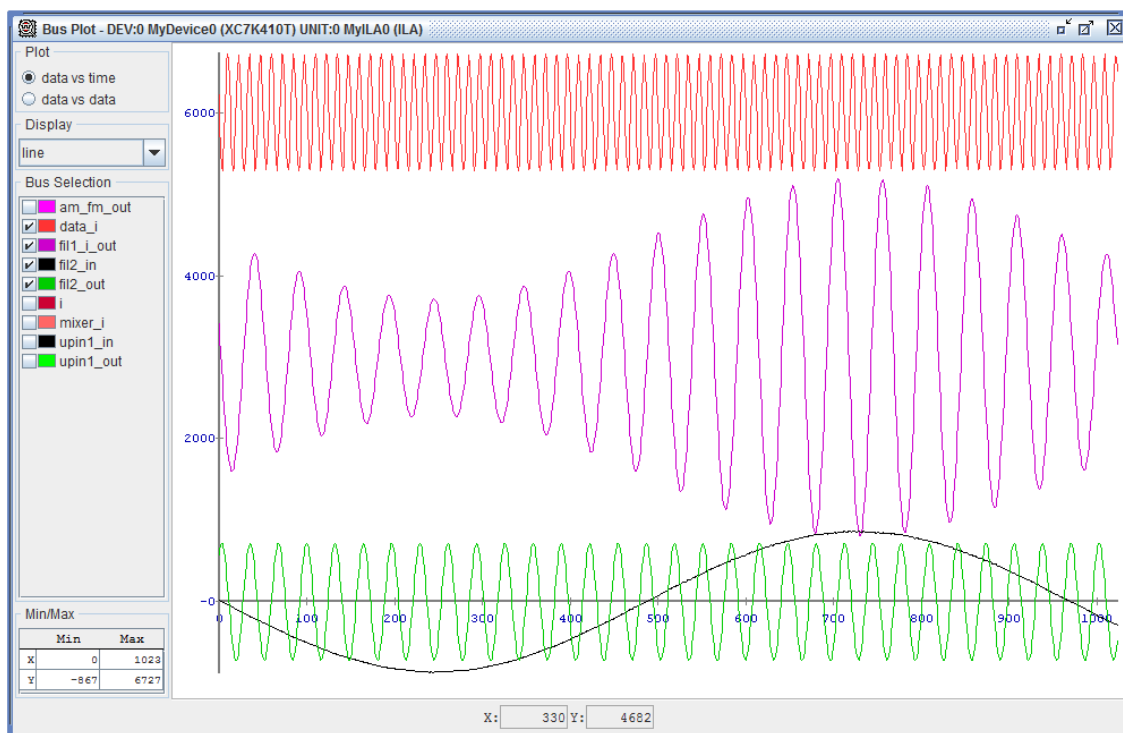
Obr. 45: Vyčítání registrů po USB 3.0 pomocí aplikace Control Center

2.5.2 Úzkopásmový režim

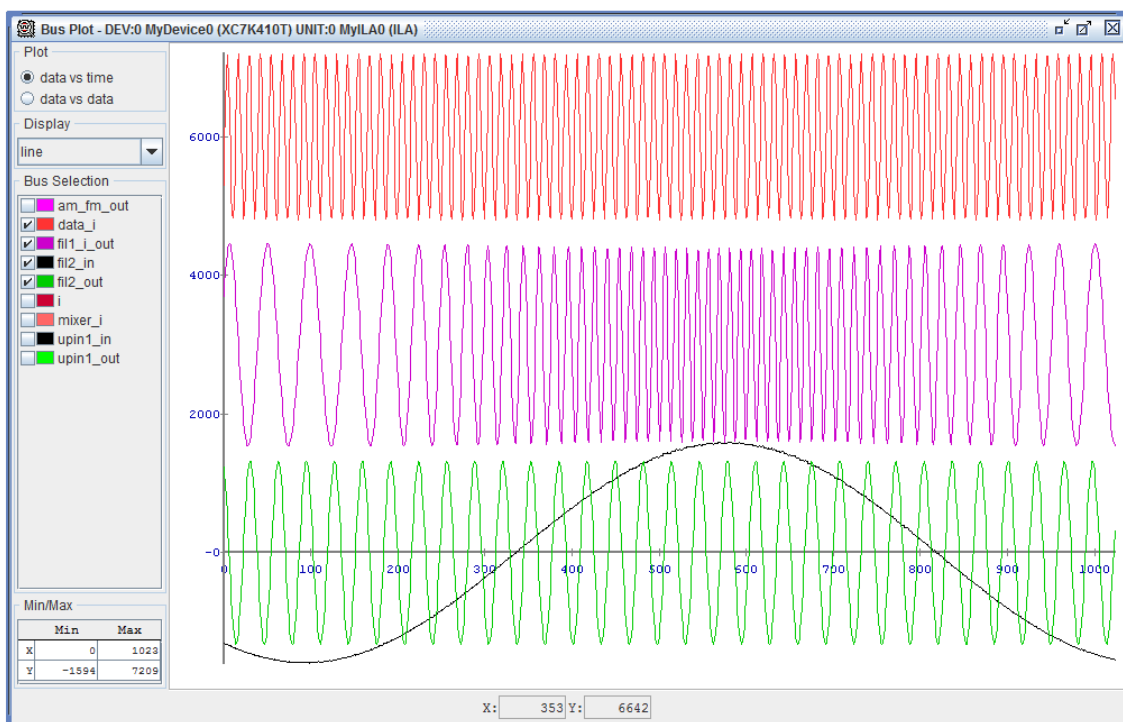
Na dvou následujících obrázcích jsou zobrazeny průběhy pořízené z analyzátoru Chipscope. První obrázek zobrazuje demodulaci signálu modulovaného amplitudově a druhý kmitočtově. Zobrazeny jsou celkem 4 průběhy. Nosný kmitočet byl na generátoru nastaven na 20 MHz a modulačním signálem byla funkce sinus o kmitočtu 1 kHz.

Červený průběh odpovídá vzorkovanému signálu z AD převodníku. Fialový průběh zobrazuje imaginární složku za první filtrací. Tedy za CIC a FIR filtrem, jedná se již o směřovaný signál se sníženým vzorkovacím kmitočtem 256x. Černá křivka zobrazuje již demodulovaný signál a zelený signál je pořízen za posledním decimačním filtrem. Tedy výstup již do počítače o vzorkovacím kmitočtu 32,5 kHz.

Na první pohled můžou zarazit měřítka pořízených charakteristik, případně DC offset. Měřítka i offset byly upraveny tak, aby se signály dobře zobrazovaly. U signálů se sníženým vzorkovacím kmitočtem byla data nabrána přes fifo a poté zobrazena.



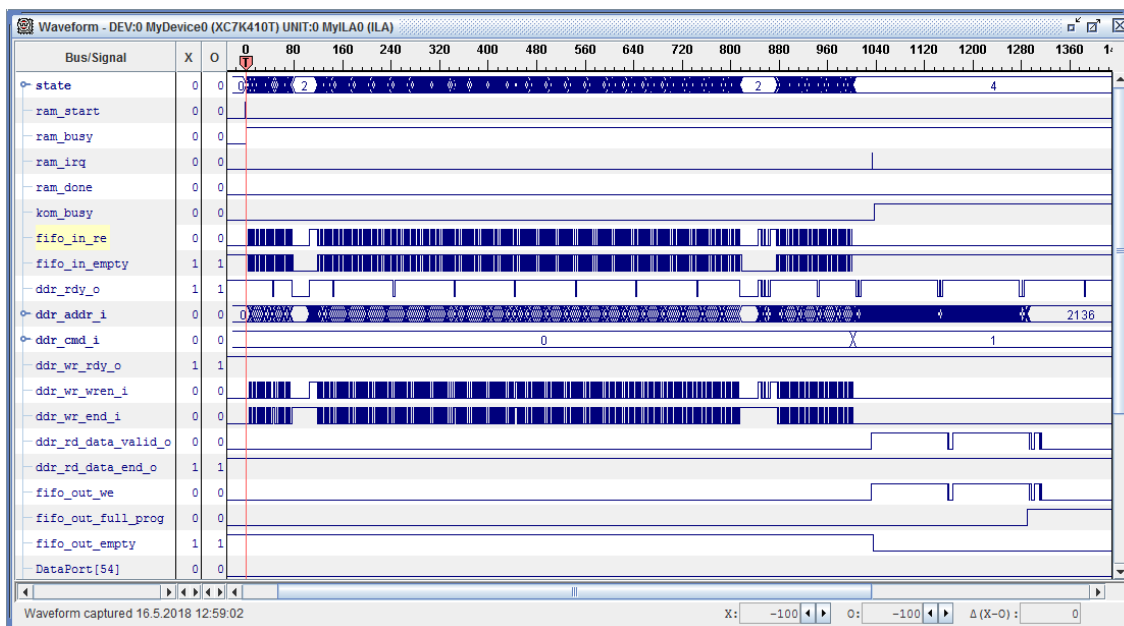
Obr. 46: Analýzátor chiscopce - demodulace AM



Obr. 47: Analýzátor chiscopce - demodulace FM

2.5.3 RAM paměť

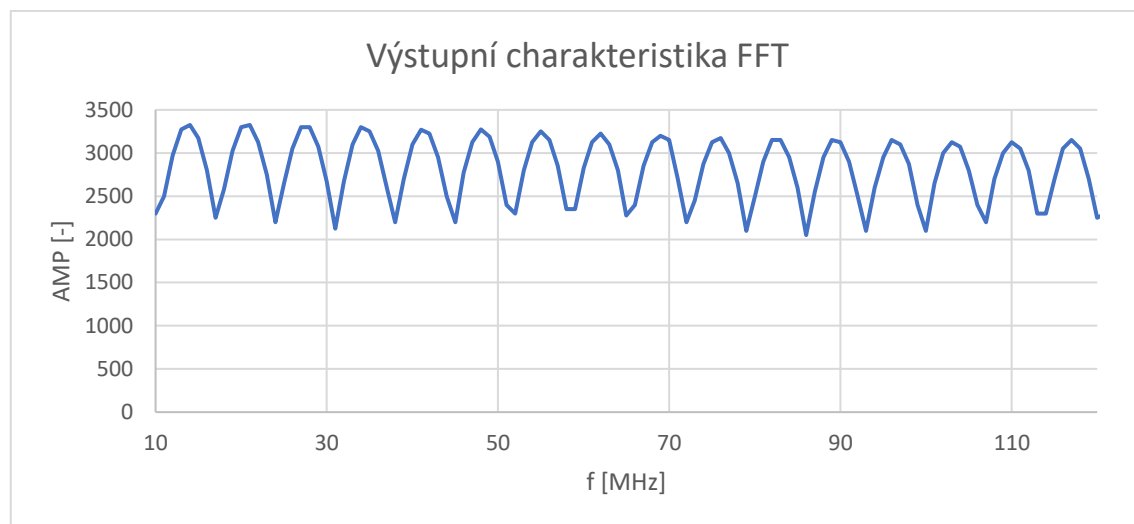
Na obrázku pod odstavcem jsou vidět časové průběhy pořízené pomocí analyzátoru Chiscope. Jedná se o signály sloužící pro obsluhu MIG jádra, které zajišťuje obsluhu DDR paměti. MIG jádro je řízeno stavovým automatem, který je popsán výše. V analyzátoru mu odpovídá signál `state`. Trigger Chipscope byl nastaven na hodnotu log. 1 signálu `ram_start`, který pochází z počítačové aplikace a slouží pro spuštění záznamu. Následně dochází k vyčítání vstupního fifa, pokud jsou v něm zapsána data (signály `fifo_in_re` a `fifo_in_empty`). Souběžně s vyčítáním nastává zápis do DDR paměti pomocí signálů `ddr_wr_wren_i` a `ddr_wr_end_i`. Během zápisu se kontrolují stavy paměti `ddr_rdy_o` a `ddr_wr_rdy_o`, zda je možný zápis. Se zápisem se také inkrementuje adresa pro zapisovaná data `ddr_addr_i` a příkaz pro zápis se zadává signálu `add_cmd_i`, kdy zápisu odpovídá dekadicky hodnota 0. Po zápisu odpovídajícího počtu dat se stavový automat přepíná do stavu pro vyčítání RAM (`state = 4`). Přepíná se signál `add_cmd_i` na hodnotu 1, což odpovídá čtení. Správné vyčtení dat signalizuje signál `ddr_rd_data_valid_o` a s tímto signálem dochází k zápisu do výstupního fifa. Odesílá se příznak IRQ a kontroluje se počet dat ve fifu pomocí signálu `fifo_out_full_prog`. Při výčtu dat USB blokem dochází k poklesu zapsaných dat ve fifu a zmíněný signál změní svoji hodnotu na log. 0, tímto opět dochází k vyčítání RAM, dokud se paměť nevyčte celá.



Obr. 48: Časové průběhy obsluhy RAM paměti

2.5.4 Spektrum

Při testování jádra FFT se proměřila výstupní charakteristika v závislosti na kmitočtu. Zvolená amplituda signálu na generátoru byla -30 dBm. Výsledná nelinearita je vidět v následujícím grafu. Pro reálné využití tohoto režimu by bylo potřeba proměřit nelinearitu přesněji a do aplikace implementovat korekce.



Obr. 49: Nelineární charakteristika FFT

2.6 Navržená aplikace

Na následujících obrázcích bude zobrazena aplikace za použití v různých režimech. Jednotlivé režimy se přepínají v záložkách a obsahují veškerá nastavení, případně spouštění.

2.6.1 Záložka Spectrum

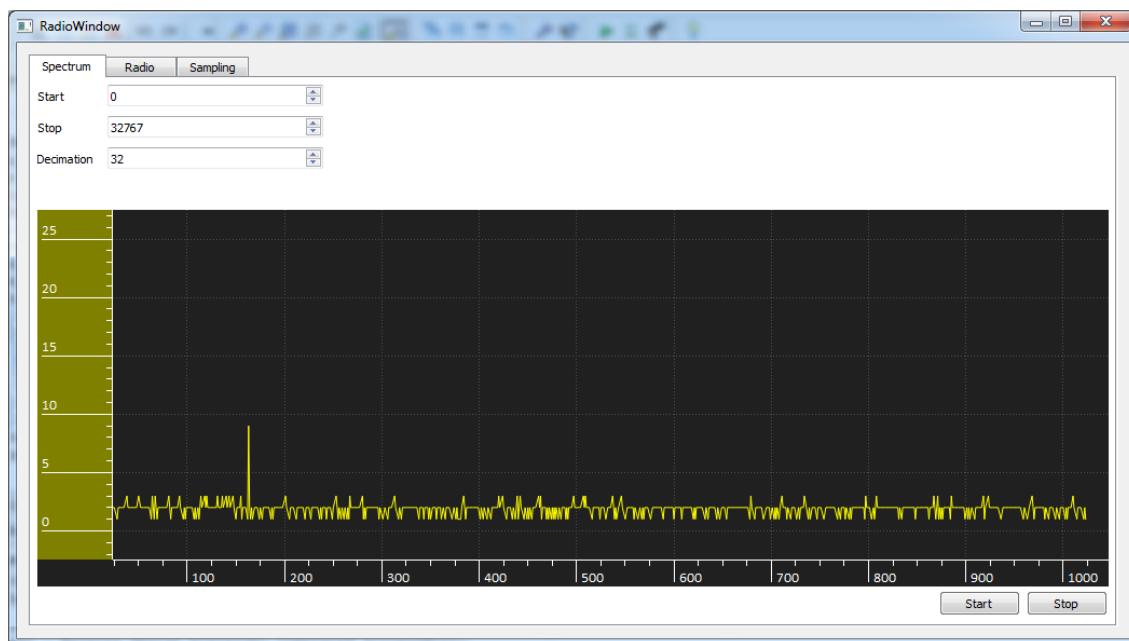
Prvním režimem je režim spektra, kdy byl použit signál o kmitočtu 20 MHz a síle signálu -80 dBm, na druhém obrázku je použit signál s amplitudou -60 dBm. Tyto hodnoty byly použity pro ukázkou citlivosti navrženého zařízení. Při náhledu na osu x si lze všimnout hodnot, osa totiž není přepočítána na frekvenci, ale zobrazuje čistě index příchozích vzorků. Osa y také nezobrazuje vhodné jednotky (dBm, mV), ale přímo hodnoty získané z FFT. Důvodem je, že se jedná pouze o testovací verzi, hlavním cílem bylo data přijmout a zobrazit. Výslednou frekvenci lze vypočítat podle vzorce:

$$f = \frac{250000000}{65536} \cdot dec \cdot i \quad (20)$$

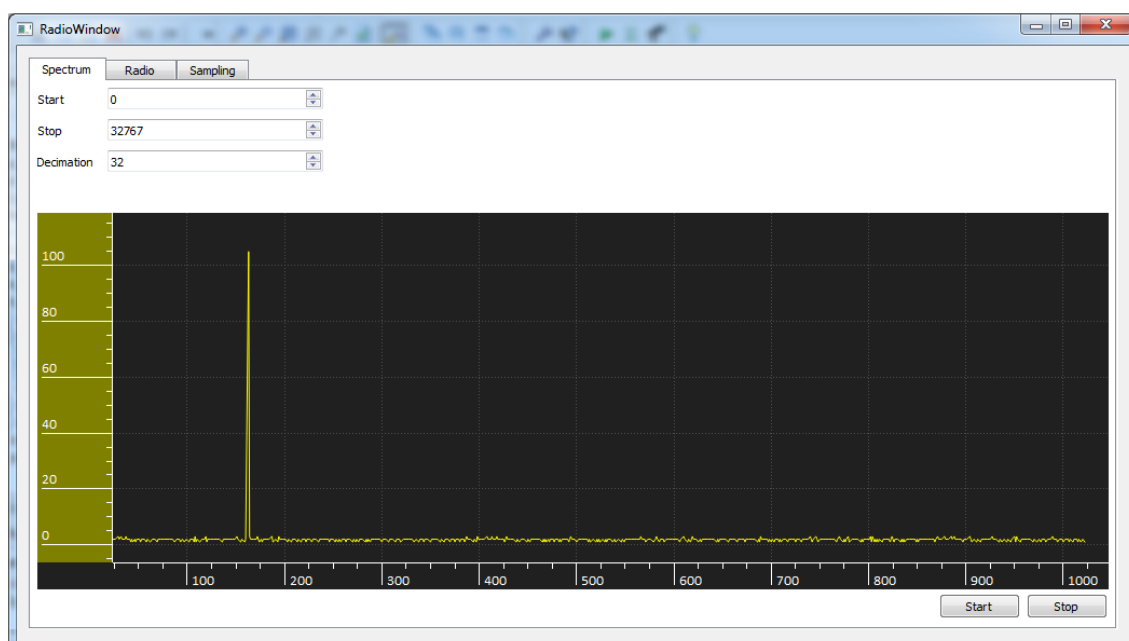
dec ... použitá decimace

i ... index vzorku v grafu

Zobrazená záložka slouží pro řízení režimu spektra. Obsahuje vstupní pole pro zobrazovaný počáteční index, koncový index a hodnotu pro decimaci. Spouští se tlačítkem Start a ukončuje tlačítkem Stop.



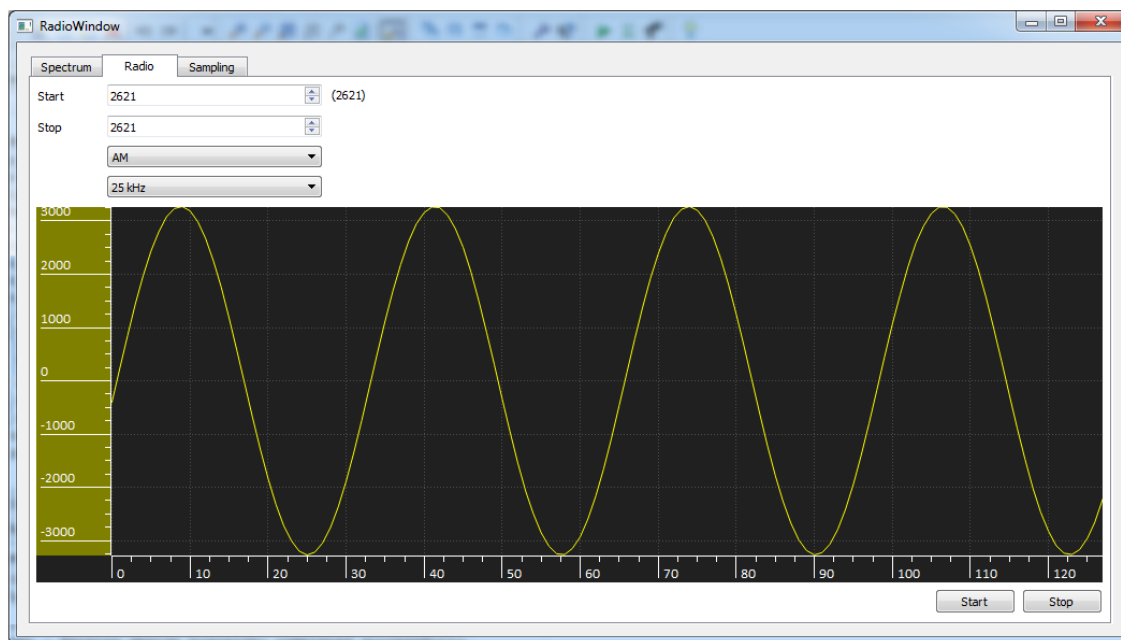
Obr. 50: Aplikace - režim Spectrum, signál sinus 200 MHz -80 dBm



Obr. 51: Aplikace - režim Spectrum, signál sinus 200 MHz -60 dBm

2.6.2 Záložka Radio

Nyní následuje záložka pro úzkopásmový režim, řízení má hodně podobné jako záložka pro spektrum. Hodnota v závorce za prvním vstupním polem ukazuje spektrální čáru, kde je zrovna zařízení naladěno. Vstupní hodnoty jsou ve spektrálních čarách. Pro ukázkou byl použit signál o nosném kmitočtu 10 MHz a modulační signál byla funkce sinus o kmitočtu 1 kHz s amplitudovou modulací.

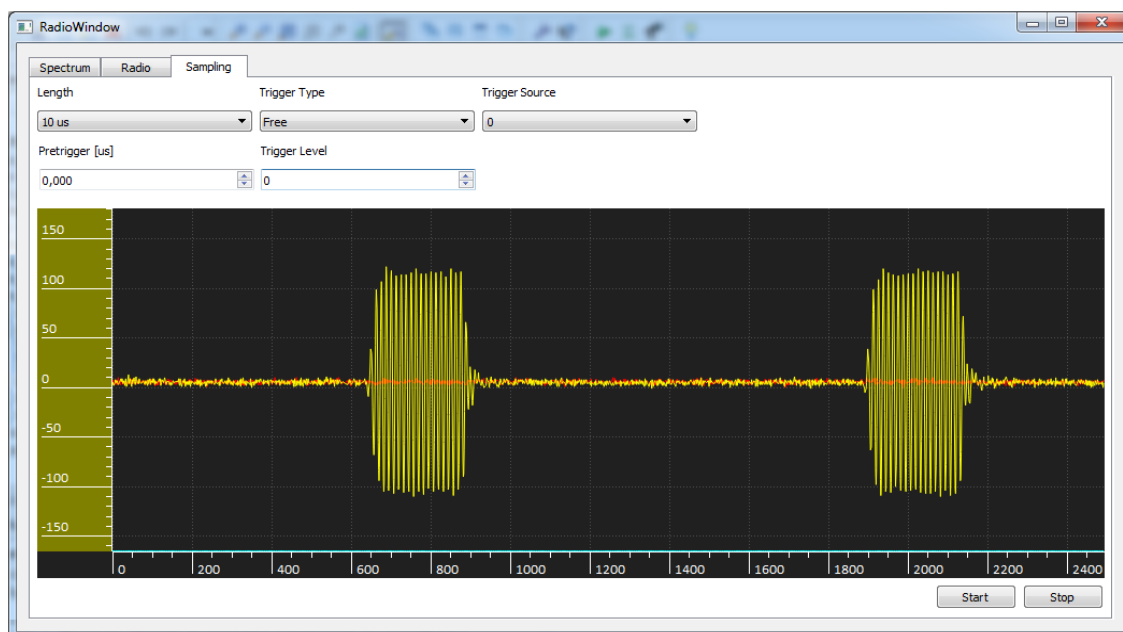


Obr. 52: Aplikace - režim úzkopásmový

2.6.3 Záložka Sampling

Poslední záložkou je záložka pro vzorkovací režim neboli sample režim. Opět obsahuje několik vstupních polí. Délka záznamu, typ triggeru, kanál a level pro trigger nebo délku předstihu v mikrosekundách. Opět jsou zde tlačítka Start a Stop pro spuštění záznamu. Teoreticky tlačítko Stop není potřeba, protože záznam není kontinuální, proběhne pouze jednou. Ovšem je takto možné kdykoliv přerušit přenos dat.

Na záznamu lze vidět dva signály. Na první převodník byl přiveden pulzně modulovaný signál. Jednalo se o pulzy s periodou 5 μ s, délkou 1 μ s a kmitočtem 20 MHz (žlutá charakteristika). Druhý převodník byl bez signálu a na pozadí je vidět šum (červená charakteristika).



Obr. 53: Aplikace - režim vzorkovací

3 ZÁVĚR

Cílem práce bylo navrhnout a realizovat spektrální analyzátor s funkcí záznamu surových vzorků do DDR paměti, rozšířený o úzkopásmový přijímač. Práce se dělí na dvě části. V první části se seznamuji s danou problematikou. Popisuji paměti RAM, komunikaci po USB 3.0 nebo filtrace a zpracování signálu. Ve druhé části se zabývám návrhem zařízení. Popisuji jak celý systém, tak i jednotlivé dílčí bloky. Práce obsahuje bloková schémata, stavové automaty a jejich popis. Dále obsahuje navržený komunikační protokol. Pro řízení celého systému byl vyhrazen paměťový prostor pro řídicí registry, kterými se nastavují a spouští jednotlivé režimy. Cílem byly tři funkční režimy, konkrétně vzorkovací Sample režim, přehled frekvenčního spektra a úzkopásmový konvertor s možností demodulace.

Vzorkovací režim funguje bez problémů, vyzkoušen byl trigger i pretrigger. Možností je volit z různých délek záznamu. U režimu přehled spektra má spektrální čára rozlišení 3,8 kHz při využití největšího FFT, které je dostupné jako IP jádro. U úzkopásmového konvertoru je přesnost ladění zvolena také na 3,8 kHz. A to z důvodu funkce automatického vyhledávání, které vyhledává právě ve spektru.

Velkým problémem, který se vyskytnul při implementaci se ukázalo USB 3.0. Problém je v ovladači dodávaném výrobcem, kdy se zařízení hlásí jako verze 2.0, a maximální přenosová rychlost byla okolo 4 MB/s. Což by při vyčítání celé RAM paměti značně zařízení omezovalo. Možné východisko vidím spíše ve využití Gigabitového ethernetu než v USB.

Co se týče aplikace, jedná se zatím o testovací verzi a pár věcí by bylo potřeba dodělat. Cílem ovšem bylo navrhnout jednoduchou aplikaci pro ovládání zařízení a zobrazení dat. Prvním nedostatkem jsou osy, které nezobrazují správné jednotky. Například osa x zobrazuje pouze index příchozích vzorků. Druhým je zadávání frekvence, které se zapisují ve spektrálních čarách a ne v jednotkách Hz.

Značné omezení je dále ve frekvenčním rozsahu zařízení. Při využití AD převodníků na vzorkovacím kmitočtu 250 MS/s se kvůli aliasingu dá použít pouze frekvenční pásmo do 125 MHz. Další nevýhodou je, že bez použití zesilovačů v analogové části je citlivost zařízení stejná jako citlivost převodníků. Tyto problémy řeší druhá deska, která již existuje a obsahuje jak směšovače, tak i zesilovače. Je možné ji připojit k použitému vývojovému kitu, ale její využití je z časových důvodů plánováno zatím do budoucna.

LITERATURA

- [1] Programovatelná logika II: FPGA. In: *AbcLinuxu* [online]. 2013 [cit. 2017-10-02]. Dostupné z: http://www.abclinuxu.cz/blog/digital_design/2013/1/programovatelná-logika-ii-fpga
- [2] Logic block. In: *Wikipedia: The Free Encyclopedia* [online]. [cit. 2017-10-02]. Dostupné z: https://en.wikipedia.org/wiki/Logic_block
- [3] What is the difference between SDRAM, DDR1, DDR2, DDR3 and DDR4? In: *Transcend*. [online]. [cit. 2017-10-04]. Dostupné z: <https://www.transcend-info.com/Support/FAQ-296>
- [4] Praktická elektronika/Spektrum signálu. In: *WIKIKNIHY: Myslete svobodně. Učte se svobodně.* [online]. [cit. 2017-10-11]. Dostupné z: https://cs.wikibooks.org/wiki/Praktick%C3%A1_elektronika/Spektrum_sign%C3%A1lu
- [5] DDR3: 8-bit. In: *EXPreview: Extreme Performance*. [online]. 2007 [cit. 2017-10-08]. Dostupné z: <http://www.expreview.com/168-5.html>
- [6] <http://www.expreview.com/168-5.html>
- [7] 5.4 FILTRACE SIGNÁLU. *Odbor termomechaniky a techniky prostředí, Fakulta strojního inženýrství, Vysoké učení technické v Brně*. [online]. [cit. 2017-10-21]. Dostupné z: <http://ottp.fme.vutbr.cz/skripta/vlab/daq/Ka05-04.htm>
- [8] NOVOTNÝ, Jan. Úvod do číslicového zpracování signálů - UCZ. *ČVUT*. [online]. 2001 [cit. 2017-10-21]. Dostupné z: <http://noel.feld.cvut.cz/vyu/ucz/cv4/>
- [9] KONÍČEK, Cyril. *Návrh a realizace akustické kamery*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky, 2013. 63 s. Vedoucí práce byl doc. Ing. Lukáš Fajcik, Ph.D.
- [10] KONÍČEK, C. *Zpracování signálu z digitálních mikrofونů typu MEMS*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 37 s. Vedoucí bakalářské práce doc. Ing. Lukáš Fajcik, Ph.D.
- [11] *Číslicová filtrace*. [online]. [cit. 2017-10-21]. Dostupné z: http://www.kiv.zcu.cz/~mautner/Azs/Azs7_Cisliciva_filtrace_zaklady.pdf
- [12] Micro-USB: USB2, USB3 and Stroom Live. In: *Stroot Live: The remote Live Streaming Solution*. [online]. 2017 [cit. 2017-10-18]. Dostupné z: <https://stroom.live/blog/index.php/tag/micro-usb/>
- [13] HAIJIAO Fan. Quickly Implement JESD204B on a Xilinx FPGA. In: *AnalogDialogue*. [online]. 2015 [cit. 2017-10-13]. Dostupné z: <http://www.analog.com/en/analog-dialogue/articles/quickly-implement-jesd204b.html>
- [14] KATAKKAR Tanaya. Field Programmable Gate Arrays (FPGA). In: *EngineersGarage*. [online]. [cit. 2017-10-02]. Dostupné z: <https://www.engineersgarage.com/articles/field-programmable-gate-arrays-fpga>
- [15] VYKYDAL, M. *Zpracování signálu z digitálního mikrofونu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 71 s. Vedoucí diplomové práce byl Ing. Zdeněk Havránek, Ph.D.
- [16] TRTÍLEK, J. *Přídavný paměťový modul pro vysokorychlostní kameru*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav mikroelektroniky (elektrotechnologie), 2017. 75 s., 28 s. příloh. Diplomová práce. Vedoucí práce: prof. Ing. Vladislav Musil, CSc.

- [17] DDR3 SDRAM. In: *Wikipedia: The Free Encyclopedia* [online]. [cit. 2017-10-04]. Dostupné z: https://en.wikipedia.org/wiki/DDR3_SDRAM
- [18] Fast Fourier Transform. In: *Wikipedia: The Free Encyclopedia* [online]. [cit. 2017-10-08]. Dostupné z: https://en.wikipedia.org/wiki/Fast_Fourier_transform
- [19] DVOŘÁK, V. Implementace výpočtu FFT v obvodech FPGA a ASIC. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 71 s. Vedoucí diplomové práce doc. Ing. Lukáš Fucik, Ph.D.
- [20] STROUHAL, A. Úzkopásmové filtry pro EKG signály. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 42 s. Vedoucí bakalářské práce doc. Ing. Jiří Kozumplík, CSc.
- [21] ŠRÁMEK, P. Implementace softwarového rádia do FPGA. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 95 s. Vedoucí diplomové práce doc. Ing. Aleš Prokeš, Ph.D.
- [22] Digital down converter. In: *Wikipedia: The Free Encyclopedia* [online]. [cit. 2017-11-21]. Dostupné z: https://en.wikipedia.org/wiki/Digital_down_converter
- [23] *Digitální zpracování signálů – Fourierova transformace, FFT, Frekvenční analýza.* [online]. [cit. 2017-10-22]. Dostupné z: https://www.fs.vsb.cz/export/sites/fs/330/.content/files/EXPME_3.pdf
- [24] PIŠI, D. Digitální paměťový osciloskop. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 99 s. Vedoucí semestrální práce Ing. Zdeněk Havránek.
- [25] USB. In: *Wikipedia: The Free Encyclopedia* [online]. [cit. 2017-10-18]. Dostupné z: <https://en.wikipedia.org/wiki/USB>
- [26] OLŠAN, Jan. USB 3.2 se představuje. Zrychlí na 20 Gb/s i se starými kabely, mohlo by přijít už brzy. In: *cnews.cz* [online]. 2017 [cit. 2017-10-18]. Dostupné z: <https://www.cnews.cz/usb-3-2-se-predstavuje-zrychli-na-20-gbs-se-starymi-kabely-mohlo-prijit-uz-brzy/>
- [27] Mercury KX1 Kintex-7 Module with USB 3.0. *XILINX: ALL PROGRAMMABLE*. [online]. [cit. 2017-11-28]. Dostupné z: <https://www.xilinx.com/products/boards-and-kits/1-2kzgy0.html>
- [28] Kintex-7 Product Advantage. *XILINX: ALL PROGRAMMABLE*. [online]. [cit. 2017-11-28]. Dostupné z: <https://www.xilinx.com/products/silicon-devices/fpga/kintex-7.html#productAdvantages>
- [29] Mercury+ PE1-200/300/400. *ENCLUSTRA: FPGA SOLUTION*. [online]. [cit. 2017-11-28]. Dostupné z: <https://www.enclustra.com/en/products/base-boards/mercury-pe1-200-300-400/>
- [30] Analog Devices Inc. AD9680-LF500EBZ. *Digi-Key: ELECTRONICS*. [online]. [cit. 2017-11-28]. Dostupné z: <https://www.digikey.com/product-detail/en/analog-devices-inc/AD9680-LF500EBZ/AD9680-LF500EBZ-ND/5300403>
- [31] *14-Bit, 1.25 GSPS/1 GSPS/820 MSPS/500 MSPS JESD204B, Dual Analog-to-Digital Converter.* [online]. [cit. 2017-11-28]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD9680.pdf>
- [32] *Designing with the EZ-USB® FX3™ Slave FIFO Interface.* [online]. [cit. 2017-11-28]. Dostupné z: <http://www.cypress.com/file/136056/download>

- [33] EZ-USB-FX3 SuperSpeed USB 3.0 peripheral controller. *CYPRESS EMBEDDED IN TOMORROW*. [online]. [cit. 2017-11-28]. Dostupné z: <http://www.cypress.com/products/ez-usb-fx3-superspeed-usb-30-peripheral-controller>
- [34] Fundamentals of I/Q Signals. In: *circuit cellar* [online]. 2015 [cit. 2018-3-14]. Dostupné z: <http://circuitcellar.com/cc-blog/fundamentals-of-iq-signals/>
- [35] *Analogové modulace*. [online]. [cit. 2018-3-15]. Dostupné z: <https://ucitel.sps-prosek.cz/~prochap/telekomunikace/22%20-%20analogov%C3%A1%20modulace.pdf>
- [36] PROKEŠ, Aleš. Využití přímé kmitočtové syntézy v radiotechnice. *VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky*. [online]. [cit. 2018-3-10]. Dostupné z: <http://www.elektrorevue.cz/clanky/03048/index.html>
- [37] RF – Mixer. [online]. [cit. 2018-3-10]. Dostupné z: http://www.sharetechnote.com/html/RF_Mixer.html
- [38] Superheterodyne receiver. In: *Wikipedia: The Free Encyclopedia* [online]. [cit. 2018-4-02]. Dostupné z: https://en.wikipedia.org/wiki/Superheterodyne_receiver
- [39] DRESLER, Tomáš. Přímá číslicová syntéza kmitočtu. In: *vyvoj.hw.cz* [online]. [cit. 2018-04-2]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/prima-cislicova-synteza-kmitoctu-dds-direct-digital-synthesis.html>
- [40] *LogiCORE IP DDS Compiler v4.0*. [online]. [cit. 2018-04-2]. Dostupné z: https://www.xilinx.com/support/documentation/ip_documentation/dds_ds558.pdf
- [41] Frekvenční modulace. In: *Encyklopedie fyziky*. [online]. [cit. 2018-3-10]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/1390-frekvencni-modulace>
- [42] Amplitudová modulace. In: *Encyklopedie fyziky*. [online]. [cit. 2018-3-10]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/1389-amplitudova-modulace>
- [43] SVOBODA, J. Přímý číslicový frekvenční syntezař. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2009. 41 s., 11 s. příloh. Diplomová práce. Vedoucí práce: prof. Ing. Miroslav Kasal, CSc.
- [44] SGANDURRA, Bob. Digital-Down Converter implementation, FPGAs offer new possibilities. In: *Military EMBEDDED SYSTEMS*. [online]. 2008 [cit. 2018-4-20]. Dostupné z: <http://mil-embedded.com/articles/digital-down-offer-new-possibilities/>
- [45] SÚKUPOVÁ, Lucie. Vzorkování signálu a aliasing. In: *Lucie Sůkupová* [online]. 2015 [cit. 2018-04-15]. Dostupné z: <http://www.sukupova.cz/vzorkovani-signalu-a-aliasing/>

SEZNAM OBRÁZKŮ

Obr. 1: Struktura hradlového pole FPGA [14]	11
Obr. 2: Struktura logické buňky hradlového pole [2]	12
Obr. 3: Příklad zobrazení spektra signálu složeného ze dvou sinusových průběhů [4].	12
Obr. 4: Filtry typu: a) dolní propust, b) horní propust, c) pásm. propust, d) pásm. zadrž [6]	13
Obr. 5: Struktura filtru s klouzavým průměrem [10]	14
Obr. 6: Frekvenční charakteristika filtru [11]	14
Obr. 7: Průměrování 12ti vzorků [11]	15
Obr. 8: Výsledky průměrování při kaskádním řazením filtrů [11]	15
Obr. 9: Struktura filtru CIC třetího řádu s decimačním faktorem 64 [9]	15
Obr. 10: Frekvenční odezva CIC filtru [10]	16
Obr. 11: Struktura FIR filtru [8]	17
Obr. 12: Ukázka prostředí Fdatool při návrhu FIR filtru [21]	17
Obr. 13: Princip směřování [34]	18
Obr. 14: Značka pro směřování	18
Obr. 15: Architektura syntežátoru DDS	19
Obr. 16: Základní struktura DDC	20
Obr. 17: Princip amplitudové modulace [42]	21
Obr. 18: Princip frekvenční modulace [41]	22
Obr. 19: Podvzorkování signálu [45]	23
Obr. 20: Zpracování dat pomocí DDR2 a DDR3 [5]	24
Obr. 21: Vybrané typy USB konektorů [12]	25
Obr. 22: Zvolený vývojový modul	26
Obr. 23: Architektura modulu Mercury KX1 [27]	27
Obr. 24: Modul KX1 [27]	27
Obr. 25: Vrstvy vysílací části JESD204 v AD převodníku [13]	29
Obr. 26: Blokové schéma navrženého systému	30
Obr. 27: Architektura čipu Cypress FX3 [33]	35
Obr. 28: Propojení FPGA a čipu FX3 při režimu Slave Fifo Interface [32]	35
Obr. 29: Vnitřní struktura čipu FX3 při režimu Slave Fifo Interface [32]	36

Obr. 30: Blokové schéma USB bloku.....	39
Obr. 31: Stavový automat řadiče USB.....	40
Obr. 32: Blokové schéma pro režim Spectrum.....	41
Obr. 33: Blokové schéma pro Úzkopásmový režim	42
Obr. 34: Blokové schéma DDC	43
Obr. 35: Princip ladění podle zvolené demodulace, vlevo AM, vpravo FM.....	43
Obr. 36: Navržený filtr typu DP 6,25 kHz.....	44
Obr. 37: Navržený filtr typu DP 12,5 kHz.....	45
Obr. 38: Navržený filtr typu DP 25 kHz.....	45
Obr. 39: Navržený filtr typu DP 50 kHz.....	46
Obr. 40: Navržený filtr typu DP 100 kHz.....	46
Obr. 41: Navržený filtr typu DP 200 kHz.....	47
Obr. 42: Fázový diskriminátor	47
Obr. 43: Blokové schéma Sample bloku	48
Obr. 44: Stavový automat pro řízení RAM.....	49
Obr. 45: Vyčítání registrů po USB 3.0 pomocí aplikace Control Center	51
Obr. 46: Analyzátor chiscope - demodulace AM	52
Obr. 47: Analyzátor chiscope - demodulace FM.....	52
Obr. 48: Časové průběhy obsluhy RAM paměti.....	53
Obr. 49: Nelineární charakteristika FFT	54
Obr. 50: Aplikace - režim Spectrum, signál sinus 200 MHz -80 dBm.....	55
Obr. 51: Aplikace - režim Spectrum, signál sinus 200 MHz -60 dBm.....	55
Obr. 52: Aplikace - režim úzkopásmový	56
Obr. 53: Aplikace - režim vzorkovací	57

SEZNAM TABULEK

Tab. 1: Vývoj RAM paměti [3]	23
Tab. 2: Přehled verzí USB [25]	24
Tab. 3: Povel pro řízení systému	31
Tab. 4: Přehled registrů.....	32
Tab. 5: Přehled možných délek záznamů u vzorkovacího režimu	33
Tab. 6: Rozčlenění adresného prostoru.....	34
Tab. 7: Protokol komunikace po USB	36
Tab. 8: Možné hodnoty pro kontrolní Byte	37
Tab. 9: Formát dat pro úzkopásmový režim bez demodulace	37
Tab. 10: Formát dat pro úzkopásmový režim s demodulací.....	37
Tab. 11: Odesílání dat spektra	38
Tab. 12: Odeslání příznaku u vzorkovacího režimu	38
Tab. 13: Vyčítání dat z paměti RAM.....	38
Tab. 14: Nastavení jednotlivých filtrů pro dané šířky pásma a odladění pro FM	44